Postprint of an Improved LIC Algorithm Based on Nonlinear Gradient Color Mapping

Authors: Gao Maoting, Dong Hongzhou, Zhou Fan

Date: 2018-06-19T00:00:00+00:00

Abstract

The texture generated by the Line Integral Convolution (LIC) algorithm reflects the directional structure of the entire vector field, yet fails to convey vector field magnitude. To address this limitation, we propose an improved LIC algorithm based on non-linear gradient color mapping. This method combines vector field magnitude with white noise as the input texture for LIC, employs FastLIC principles and partitions texture regions for synchronous LIC execution to enhance computational efficiency. Subsequently, the vector field magnitude undergoes non-linear transformation, and the OpenCV processing engine is utilized to parallelly implement color mapping according to a gradient color mapping scheme. Finally, synthesis coefficients are determined from the LIC-generated grayscale texture and the color mapping results, an accumulation function is constructed to enhance both components, and linear synthesis operations are performed to obtain the final visualization. Experimental results from visualizing two typical vector fields–global ocean current fields and wind fields–and comparative studies with other algorithms demonstrate that the improved algorithm produces visualization results with clear texture, effectively displays both direction and magnitude of the vector field, and more accurately reflects comprehensive information and local variations within the vector field.

Full Text

Preamble

Improved LIC Algorithm Based on Nonlinear Gradual-Changing Color Mapping

Gao Maoting, Dong Hongzhou, Zhou Fan

(College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China)

Abstract: The texture generated by Line Integral Convolution (LIC) algorithms effectively reveals the directional structure of entire vector fields, yet fails to represent field intensity magnitude. To address this limitation, we propose an improved LIC algorithm based on nonlinear gradual-changing color mapping. Our approach combines vector field intensity with white noise as the input texture for LIC, employing FastLIC principles and partitioning the texture into regions for synchronous LIC computation to enhance efficiency. Vector field intensity undergoes nonlinear transformation, and OpenCV processing engines parallelize the color mapping according to a gradual-changing color scheme. Finally, synthesis coefficients are determined from the LIC-generated grayscale texture and color mapping results, with an accumulation function constructed to enhance both components before linear synthesis yields the final visualization. Comparative experiments on global ocean current and wind fields—two typical vector fields—demonstrate that our improved algorithm produces clear textures that effectively display both direction and intensity, more accurately reflecting comprehensive field information and local variations.

Keywords: line integral convolution; vector field; intensity; visualization; cumulative function

0 Introduction

Vector field visualization is crucial for understanding and analyzing motion patterns, providing foundational support for discovering characteristic field structures. However, vector data possesses both magnitude and direction attributes, and no directly applicable visualization model currently exists. Real-world vector fields such as ocean currents and wind fields exhibit complexity and particularity that traditional point icon and vector line methods can no longer adequately visualize. The Line Integral Convolution (LIC) algorithm, introduced by Cabral and Leedom at SIGGRAPH ' 93, generates continuous, smooth textures that clearly exhibit directional changes and shape characteristics, overcoming the visual clutter inherent in conventional methods.

While LIC effectively displays directional information, it cannot reflect intensity magnitude and remains computationally expensive. Numerous improved algorithms have been proposed to address these issues. Zhang et al. introduced a FastLIC algorithm based on the HLS color model, mapping vector magnitude through HLS and overlaying results onto grayscale LIC textures using OpenGL. Lu et al. proposed a contrast-based LIC algorithm that quantitatively maps intensity through image contrast before synthesis with LIC textures. Zhang et al. developed an HSV-based LIC approach mapping direction to hue and intensity to saturation, converting HSV to RGB for final color textures while leveraging FastLIC for efficiency. Zhao et al. presented a lookup table-based LIC algorithm that pre-computes streamlines and maps intensity to color values for weighted averaging with LIC grayscale values. Zhan et al. implemented a CUDA-based color-enhanced LIC algorithm for parallel execution and classified color mapping to highlight regions of interest. Ding et al. designed a parallel LIC algorithm for unsteady flow fields with streamline reuse strategies and GPU-based linear intensity mapping. Han et al. proposed a GPU-accelerated streamline-enhanced LIC algorithm with adaptive step sizes to reduce computation while improving rendering speed.

Although these improved algorithms simultaneously display direction and intensity through color mapping and linear synthesis with LIC grayscale textures, they suffer from reduced brightness, diminished contrast and visual resolution, and indistinguishable colors in regions with gradual intensity variations. Linear synthesis coefficients lack reliable determination methods—excessive weighting on grayscale textures diminishes color prominence, while insufficient weighting yields unclear textures. Consequently, we propose an improved LIC algorithm based on nonlinear gradual-changing color mapping to overcome these limitations.

1 Classic Line Integral Convolution Algorithm

Line Integral Convolution (LIC) represents vector field direction information through a convolution formulation derived from motion blur concepts. As a texture-based vector field visualization method, LIC combines the advantages of geometric shape mapping and color mapping, producing spatially continuous and smooth textures that clearly reveal directional changes and shape characteristics while overcoming the visual clutter of traditional point icons and geometric vector lines.

In LIC operations, a white noise texture matching the vector field resolution serves as input. Streamlines are generated from selected points by symmetrically extending both forward and backward along the vector field. All corresponding noise values along the streamline participate in convolution according to the kernel, with results assigned as output texture grayscale values. Assuming a local streamline is represented by (s) with arc length parameter s, input white noise texture I(x), and normalized parameter k(s), the grayscale value at position x is calculated as:

$$I(x_0) = \frac{\int_{-L}^{L} I(\sigma(s)) \cdot k(s)\, ds}{\int_{-L}^{L} k(s)\, ds}, \quad \sigma(0) = x_0$$

where k(s) is the convolution kernel function, and different kernels produce different results. The classic LIC algorithm flow is shown in [Figure 1: see original paper].

2 Improved LIC Algorithm Based on Nonlinear Gradual-Changing Color Mapping

To better display vector field direction and intensity information while obtaining high-resolution visualization and improving LIC efficiency, we propose an improved LIC algorithm based on nonlinear gradual-changing color mapping. The algorithm comprises three modules, as illustrated in [Figure 2: see original paper].

The first module improves LIC operation by combining vector field intensity with white noise to form a new input texture, integrating intensity information into the LIC-generated grayscale texture for enhanced display. To accelerate computation and reduce hardware dependency, this module adopts FastLIC principles and partitions the synthetic texture into multiple regions for synchronous LIC execution.

The second module implements nonlinear gradual-changing color mapping. Based on application scenarios, a color table is selected to formulate a nonlinear gradual-changing color mapping scheme implemented via OpenCV, overcoming the poor mapping effects of linear approaches in regions with subtle intensity variations. This module operates in parallel with the first module to improve overall speed.

The third module synthesizes the grayscale texture with the nonlinear color mapping results. Linear synthesis coefficients are determined from each point' s grayscale texture value and mapping result. An additional cumulative function transforms both components accordingly before final synthesis produces the color texture result, overcoming the drawbacks of conventional linear synthesis methods such as dark results, unclear textures, and difficult coefficient determination.

2.1 Improved LIC Operation

In classic LIC, a white noise texture T of size X×Y is generated to match the vector field dimensions. For computational convenience, we represent texture values as T(x,y) = rand(x,y), where rand(x,y) [0,1] produces random values. While some approaches color the noise to distinguish motion directions, our method integrates original vector field intensity information into the white noise texture for LIC computation.

From the original vector field, we obtain an intensity matrix M. To integrate intensity into white noise matrix T, we normalize M. Let the original intensity matrix be M with elements m__xy, and the normalized matrix be N with elements n__xy. The normalization method is:

$$n_{xy} = a + \frac{(b-a)(m_{xy} - m_{\min})}{m_{\max} - m_{\min}}$$

where m__max and m__min are the maximum and minimum values in M. Since T elements [0,1], we normalize N to [0,1] by setting a = 0, b = 1.

From the LIC equation, output texture correlates positively with input white noise. After integrating intensity, this correlation must be preserved. We therefore weightedly sum the white noise matrix T with normalized intensity matrix N. Let the weight coefficients be and , with the new texture Q = g(T,N, ,) = T + N, where + = 1. Larger emphasizes direction information, while larger highlights intensity information.

After obtaining the new input texture, LIC generates streamlines extending length L forward and backward from each point. We employ numerical integration using the variable-step fourth-order Runge-Kutta method for highest accuracy in revealing complex, fine-scale structures. The trajectory over time interval Δt is:

$$s(t_{k+1}) = s(t_k) + \int_{t_k}^{t_{k+1}} V(s(t), t)\, dt$$

where V(s(t), t) represents streamline velocity. The discrete form for point x is:

$$I(x_1) = \frac{1}{2L+1} \sum_{i=-L}^{L} T(\sigma_i)$$

However, streamline generation is computationally expensive. To improve efficiency, we adopt FastLIC principles, which exploit correlations between adjacent points. As shown in [Figure 3: see original paper], streamlines for neighboring points share substantial overlapping regions. If the grayscale value at x is known, the value at adjacent point x can be obtained without recomputing the entire streamline:

$$I(x_2) = I(x_1) - \Delta I_1 + \Delta I_2$$

where ΔI and ΔI represent the convolution results from the non-overlapping regions. For discrete sampling with n points along each streamline (2L+1 total points), using a box filter kernel ensures equal coefficients summing to 1, yielding:

$$I(x_1) = \frac{1}{2n+1} \sum_{i=-n}^{n} T(\sigma_i)$$

Building on FastLIC, we partition the input texture into n regions for synchronous execution, significantly reducing computation while maintaining quality through embedded variable-step Runge-Kutta integration.

2.2 Nonlinear Gradual-Changing Color Mapping

To better represent intensity information and improve visual quality, we implement nonlinear gradual-changing color mapping using a self-selected color table, rendered efficiently via OpenCV. OpenCV is a cross-platform computer vision library whose powerful algorithms accurately sample colors from selected tables without external dependencies, compiling efficiently for high-performance color mapping. This process runs in parallel with LIC operations to enhance overall algorithm speed.

2.2.1 Nonlinear Field Intensity Value Transformation Vector field intensity often distributes non-uniformly. Linear mapping concentrates results in narrow color ranges when intensity varies slightly, making local distinctions difficult. We address this through nonlinear transformation. From original intensity mag, we compute transformed intensity newmag using:

$$\mathrm{newmag} = \frac{\mathrm{mag}^{1/c}}{\mathrm{mag}_{\mathrm{max}}^{1/c} - \mathrm{mag}_{\mathrm{min}}^{1/c}}$$

where c is the nonlinear mapping factor (typically c = 0.1 in experiments) and mag_max is the maximum intensity. This transformation, shown in [Figure 4: see original paper], expands the mapping range for subtle intensity variations.

2.2.2 Gradual-Changing Color Mapping For ocean current and wind field visualization, we reference color tables from PanoplyWin software, a professional ocean data visualization tool released by NASA GISS. Selected color tables are shown in [Figure 5: see original paper].

After nonlinear transformation, we implement smooth color transition through gradual-changing mapping. OpenCV reads the color table size w×h. To map newmag [0,1] to the table, we scale it:

$$p = \lfloor \mathrm{newmag} \times w \rfloor, \quad p \in [0, w]$$

As illustrated in [Figure 6: see original paper], we divide the color table length into w segments. For any p falling between adjacent points p_m and p_{m+1}, with distances l and l respectively, the color value at p is:

$$\mathrm{color}(p) = \frac{l_2 \cdot \mathrm{color}(p_m) + l_1 \cdot \mathrm{color}(p_{m+1})}{l_1 + l_2}$$

This interpolation ensures smooth color gradients across the entire intensity range.

2.3 Synthesis of Grayscale Texture and Color Mapping Results

After both modules complete, synthesis produces the final visualization. Let LIC_R be the grayscale texture and color_R the mapping result. Common synthesis methods include:

$$\text{Result} = t \cdot \text{LIC\_R} + (1-t) \cdot \text{color\_R}$$

or multiplicative approaches, where t (0,1) is the synthesis coefficient. However, these methods produce dark results with reduced contrast and unclear textures.

We construct an additional cumulative distribution function for enhancement. First, determine synthesis coefficients from grayscale and mapping values. Then transform both results through cumulative functions before final synthesis. The process:

a) Find maximum and minimum grayscale values in LIC_R: I_max and I_min

b) Divide LIC_R into n equal intervals L , L , ⋯, L

c) Count pixels h_i in each interval and compute percentages: s_i = h_i / (X×Y)

d) Reassign grayscale values: I' (x,y) = 255 × Σ_{j=0}^{i} s_j for pixels in interval i

This cumulative transformation preserves the original grayscale variation trend while enhancing display quality. The final synthesis combines enhanced grayscale and color mapping results, overcoming coefficient determination difficulties and poor synthesis effects.

3 Experiments

3.1 Dataset and Experimental Environment

We selected global ocean current and wind fields as experimental subjects, using observational data from NOAA in NetCDF format with 6-hour temporal resolution and spatial resolution covering 180°W~180°E, 90°S~90°N.

The experimental environment consisted of Windows 10, Intel Core i5 processor, 8 GB RAM, VS2013 development tools, OpenCV graphics library, and C++ implementation.

3.2 Visualization Effect Comparison Experiments

Using professional color tables suitable for ocean currents and wind fields, our algorithm produces visualizations with significant color differentiation compared to HLS/HSV-based LIC approaches. To ensure fair comparison, all experiments use identical color tables, comparing results against classic LIC, linear color mapping, and Linear Color-mapping LIC (LCLIC) algorithms to validate our method' s effectiveness.

For global ocean current visualization, our improved LIC module output is shown in Figure 7: see original paper, while classic LIC results appear in Figure 7: see original paper (black regions represent land). Our approach already displays intensity distribution through 明暗渐变 (light-dark gradients) with clear directional changes.

[Figure 8: see original paper] compares nonlinear gradual-changing color mapping (a) against ordinary linear mapping (b). Our method adapts to regions with subtle intensity variations. Where linear mapping produces large uniform color patches (boxed areas in Figure 8: see original paper), our approach reveals fine intensity distributions (boxed areas in Figure 8: see original paper).

LCLIC algorithms using conventional synthesis suffer from unclear textures and undefined coefficients. [Figure 9: see original paper] shows LCLIC results with t = 0.3 (color-rich but unclear textures) and t = 0.7 (texture-dominant but dim colors), both suffering from overall darkness and weakened effects.

Our algorithm defines synthesis coefficients explicitly and constructs cumulative functions to transform both components, overcoming these limitations. [Figure 10: see original paper] presents our global current visualization results. [Figure 11: see original paper] shows global wind field visualization. Both demonstrate that our nonlinear gradual-changing color mapping LIC algorithm clearly displays directional changes and intensity information with high brightness and contrast, avoiding the color saturation reduction, darkness, and low texture discrimination of other methods while revealing local variation trends.

3.3 Algorithm Efficiency Comparison Experiments

3.3.1 Comparison with Other Visualization Algorithms To evaluate computational overhead, we compared our algorithm (NGCLIC) against point icon method (PIM), vector line method (VLM), and topology method (TM) from literature. Timing statistics are shown in .

While our algorithm consumes more time than PIM and VLM, it is faster than TM. More importantly, our results exhibit stronger correlations between texture points along flow directions, demonstrating data continuity and achieving realistic scene simulation—capabilities unattainable by PIM and VLM.

3.3.2 Comparison with Other LIC Improved Algorithms We further compared NGCLIC against HFLIC, HSVLIC, LTLIC, and NCLIC from literature [5-9]. Timing statistics appear in .

Our algorithm improves efficiency over methods in [5,7,8] while maintaining visualization quality, though it is slower than the GPU-based approach in [9].

Our speed advantages stem from FastLIC principles, partitioned synchronous computation, lightweight OpenCV engines, and parallel LIC/color mapping execution—benefits that become more pronounced with increasing data scale. However, our method requires system overhead for texture generation and module synchronization, and without GPU acceleration, it remains slower than [9]. Nevertheless, our lower hardware requirements ensure broader applicability.

4 Conclusion

We have proposed an improved LIC algorithm based on nonlinear gradual-changing color mapping to address the weak intensity representation in classic and improved LIC algorithms. Experimental validation on turbulent ocean current and wind fields demonstrates that our algorithm simultaneously displays direction and intensity with clear, accurate representation of global information and local variations. The resulting color textures are continuous and smooth, maintaining high color brightness and visual resolution for effective visualization.

Visualization of vector environments has become an urgent research priority. Our algorithm provides new methods for ocean current and wind field visualization, with potential applications to aerodynamic flow fields, magnetic fields, and other vector domains. Future work will integrate data mining techniques to explore vector field variation patterns, characteristic structures, and phenomena, establishing robust visualization-mining mechanisms.

Note: Figure translations are in progress. See original paper for figures.