

鈴木 靖子 藤代 一成 竹島 由里子†

お茶の水女子大学 理学部 情報科学科

† お茶の水女子大学 大学院人間文化研究科

1 背景と目的

1993年に Cabral と Leedom により提案された Line Integral Convolution (LIC)[1]は、入力テクスチャの各ピクセルを局所流線にそってぼかすことにより、密な流れ場の可視化を可能にした。このアルゴリズムは、シンプルな手続きにもかかわらず流れの精緻な構造や大域的な特徴を直感的かつ効果的に可視化できるため、現在、流れの可視化の代表的な技法になりつつある。この手法の拡張は多くの研究者達によって盛んに試みられており、その重要な方向性の一つに3次元化があげられる。

本稿では、任意方向の断面上に2次元流れ場のLICテクスチャを対話的にマッピングするシステムを開発し、その効果を実データの可視化実験を通じて検証する。

2 LIC アルゴリズム [1]

LIC法は、テクスチャにぼかし効果を与えることにより流れを表現する手法で、そのアルゴリズムはきわめて簡潔である。

はじめに2次元の直交格子にベクトル場を与え、各セルに対し一定の長さの流線を生成する。図1にセル

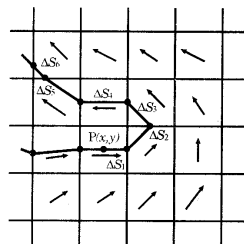


図1: 流線

$P(x, y)$ における流線を示す。 $P(x, y)$ の中心を始点と

Interactive Visualization of 3D Flow Fields Using LIC Texture Mapping

Yasuko Suzuki, Issei Fujishiro, Yuriko Takeshima†

Department of Information Sciences, Faculty of Science,

†Graduate School of Humanities and Sciences,

Ochanomizu University,

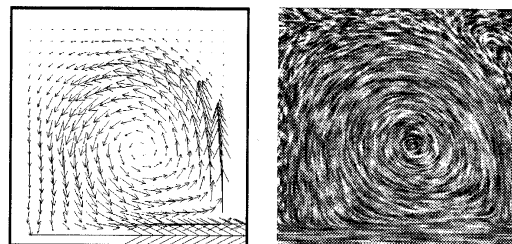
2-1-1 Otsuka, Bunkyo-Ku, Tokyo 112-8610, Japan.

し、各セルのベクトルにそってまず正方向に折れ線を定義していく。このとき、始点からの折れ線の長さ ΔS が定数 l 以下になるようにセル数 n を定める。ここで ΔS_i はセル i での折れ線の長さとする:

$$\Delta S = \Delta S_1 + \Delta S_2 + \Delta S_3 + \dots + \Delta S_n \quad (\Delta S \leq l)$$

次に、各 ΔS_i を重みとして n 個のピクセルの値(色)を対応する大きさのテクスチャ画像から抽出する。同様に負方向についても、 $P(x, y)$ の中心を始点として m 個のピクセル値を抽出し、 $n+m$ 個のピクセル値の重み和を計算して $P(x, y)$ における出力ピクセル値とする。以上の操作をすべてのピクセルについて行うことで、流線にそったぼかしの効果が得られる。さらに、セルに与えられたベクトルの大きさに応じて l の値を変化させることにより得られるぼかしの程度によって流速を表現することもできる。

図2に矢印グリフにより流れ場を表現する arrow plot法とLIC法の比較実験を行った結果を示す。入力ベクトル場に cavity flowの数值データを使用し、入力テクスチャにはホワイトノイズ画像を与えた。



(a) arrow plot 法

(b) LIC 法

図2: cavity flow データの可視化

1次渦と右上の小さな2次渦では、速度に大幅な相違が存在するので、arrow plot法では2次渦を全く観測することができない。それに対しLIC法では2次渦が明確に表現されている。なお、ここでは速度 v を対数関数によって変換し、 l の値を

$$l = a \log_b v \quad (a, b \text{ は適当な定数})$$

とすることによりダイナミックレンジを圧縮した。

3 3次元流れ場への応用

3次元流れ場を可視化できるようにLIC法を拡張する研究はさかに行われている。2次元と同様の原理で3次元流線にそってボクセル値をぼかした後にボリュームレンダリングを行う方法は、視線方向にリサンプル値を積分することにより、得られた相関テクスチャを相殺してしまうため、効果的ではないとされてきた。その代わりにとられたアプローチが3次元の物体の表面にLICテクスチャを張り付ける手法[2]である。しかし物体の表面にLICテクスチャをマッピングするとテクスチャに歪みが生じ、元の流れ場の誤った認識を引き起こしてしまう恐れがある。この問題はソリッドテクスチャリングを用いる手法[4]により解決されたが、この手法は大量の計算時間が必要とされるため、対話的な可視化に適しているとはいえない。

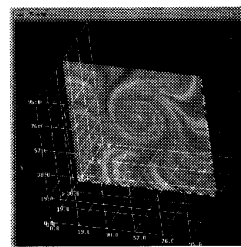
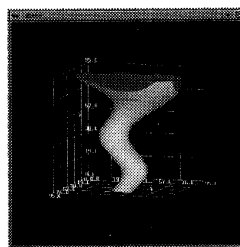
そこで本稿では、3次元のベクトル場から断面を切り出し、断面上でLICアルゴリズムを実行するアプローチを採用する。3次元空間内の任意断面上へ生成されたLICテクスチャをマッピングする方法の有効性は、すでに文献[3]で確かめられている。この手法は2次元のLICテクスチャ計算をそのまま適用でき、3次元へ適用する際に生じる歪みの補正等の複雑な処理を必要としない。対話的に断面の位置や向きを変化させ、空間内部の流れ場を連続的にさまざまな角度から描写することにより、流れ場の内部構造の特徴を感覚的に捉えることが可能となる。

4 実験と考察

流速の等値面表示と任意断面上へのLICテクスチャマッピングとの比較実験を行った。可視化環境にはAVS/Express Ver5.0¹を使用した。LIC法はAVS/Expressのモジュールとして実装され、既存のslice planeモジュールと組み合わせることで対話的な可視化を可能にしている。

図3に竜巻データ(48×48×48)の可視化結果を示す。等値面表示(図3(a))から、竜巻が水平方向に螺旋を描きながら上方へと伸びていることがわかるが、これだけではある流速値の分布しか一度に捕らえることはできない。一方LICテクスチャマッピング図3(b)を用いて平行な面を切り出した結果からは中心に向かって流れに対し外側に向かって逆方向の流れが発生していることがわかる。さらに断面の高さを任意に変化させることにより、竜巻の内部構造の全体像をつかむことができる。実際、垂直方向に向かって渦の大きさが次第に大きくなっていく様子が容易に観測される。

¹AVS/ExpressはAdvanced Visual Systems, Inc.の登録商標である。



(a) 流速の等値面表示 (b) LICテクスチャマッピング
図3: 竜巻データの可視化

5 まとめと今後の課題

3次元の流れ場から任意に切り出された断面上へLICテクスチャを張り付け、切り出す断面を対話的に変化させることにより3次元の流れ場の内部を表現する方式の有効性が再確認できた。

今後は区間型ボリューム[5]によって定義される任意の3次元サーフェスへのマッピング、そして処理アルゴリズム全体の並列化を行う予定である。

謝辞：第4節で使用した竜巻データを提供してくれた米国カリフォルニア州立大学デイビス校のN. Max教授に深謝する。

参考文献

- [1] B. Cabral and L. Leedom: "Imaging Vector Fields Using Line Integral Convolution," In *Computer Graphics Proceedings, Annual Conference Series*, August 1993, ACM SIGGRAPH, pp.263-270.
- [2] D. Stalling: "LIC on Surfaces," In *Texture Synthesis with Line Integral Convolution*, K. -L. Ma (Organizer), ACM SIGGRAPH Course Notes 8, August 1997, pp.51-62.
- [3] C. Rezk-Salama, P. Hastreiter, C. Teitzel, and T. Ertl: "Interactive Exploration of Line Integral Convolution Based on 3D-Texture Mapping," In *Proceedings of IEEE Visualization'99*, San Francisco, October 1999, pp.233-240.
- [4] X. Mao, M. Kikukawa, N. Fujita, and A. Imamiya: "Line Integral Convolution for Arbitrary 3D Surface through Solid Texturing," In *Proc. 8th Eurographics Workshop on Visualization in Scientific Computing*, W. Lefer and M. Grave (eds.), Springer-Verlag, 1997, pp. 57-69.
- [5] I. Fujishiro, Y. Maeda, H. Sato, and Y. Takeshima: "Volumetric Data Exploration Using Interval Volume," *IEEE Transactions on Visualization and Computer Graphics*, vol. 2, no. 2, pp. 144-155, June 1996.