グラフィクスとCAD 108－12
（２００２．８．８）

ボリュームグラフィックス（VG）クラスタによる 3D LIC レンダリングの並列化

村木　茂
(独)産業技術総合研究所
ボリュームグラフィックス連携研究体

鈴木　靖子
三菱電機（株）
情報技術総合研究所

藤代　一成
お茶の水女子大学大学院
人間文化研究科

あらまし
我々は 3D Line Integral Convolution (3D LIC) 法のボリュームレンダリングが 3 次元ベクトル場の可視化方法として有望であると考えている．本稿では我々が共同で開発中の高並列計算可視化システム「VG クラスタ」に対話的な 3D LIC アニメーション機能を実装し，3 次元流の可視化処理としての有効性を検討する．

Parallel 3D LIC Rendering by the Volume Graphics (VG) Cluster

Shigeru Muraki
Collaborative Research Team of Volume Graphics, AIST

Yasuko Suzuki
Information Technology R&D Center, Mitsubishi Electric Corp.

Issei Fujishiro
Graduate School of Humanities and Sciences, Ochanomizu Univ.

Abstract
Volume rendering of three-dimensional Line Integral Convolution (3D LIC) is a promising visualization technique for 3D vector fields. In this paper, we implement an interactive 3D LIC animation function on a Volume Graphics (VG) cluster prototype system, which is in the development stage in our collaborative research, and discuss the effectiveness for the 3D flow visualization.

1. はじめに

ホワイトノイズテクスチャを流線方向にぼかす Line Integral Convolution （LIC）法[1]は，現在最も注目されている 2 次元流の可視化技法である．これを 3 次元に拡張し（3D LIC），ボリュームレンダリングで画面に投影することにより, 流れの 3 次元構造が映像化できると考えるのは自然なことであろう．しかし 3D LIC の単純なボリュームレンダリングは，3 次元流線の濃淡値を奥行き方向に重ねてしまうため, 流れの可視化手段としてあまり効果的でないと考えられていた[2,3]. 我々は十分に高速なボリュームレンダリングが可能であれば, 位相シフトアニメーション法により動画像化した 3D LIC 時系列に対して，対話的に視点位置や伝達関数を変更しながらボリュームレンダリングを行うことで, 流れの 3 次元構造が容易に把握できると考えている．

現在我々が開発中のボリュームグラフィックス（VG）クラスタ[4]はこうした用途に適した高並列計算可視化システムである. 本稿では VG クラスタプロトタイプ上に，3 次元ベクトル場からの 3D LIC 時系列生成処理と，対話的な動ボリュームデータ可視化処理を並列に実装し，検討を行ったので報告する．

2. 3D LIC 法

2.1. LIC 法

LIC 法はホワイトノイズテクスチャとベクトル場データを入力し, 流線に沿ってテクスチャをぼかしたような画像を生成する流れの可視化法である．オリジナルの 2 次元 LIC アルゴリズム[1]では，図１のような 2 次元直交格子にベクトル場を与え, ある一つのセルの中心 $P(x, y)$ を始点として各セルのベクトルにそって一定の長さの折れ線状の局所流線を生成する．このとき，まず $P(x, y)$ から正方向に折

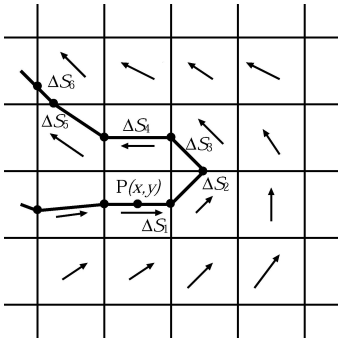

図 1 LIC 法における局所流線の定義

れ線の長さ

$$\Delta S = \sum_{i=1}^{n} \Delta S_i$$

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 を横切る折れ線の長さとする．同様に負方向についても，$P(x,y)$ の中心を始点として m 個の画素値を抽出する．次に，各 ΔS_i を重みとして $n+m$ 個の画素値（色）を対応する大きさのホワイトノイズテクスチャから抽出し，$n+m$ 個の画素値の重み和を計算して $P(x,y)$ における出力画素値

$$F_{\text{out}}(x,y) = \frac{\sum_{i=0}^{n} F_{\text{in}}\left(\lfloor P_i \rfloor\right) h_i + \sum_{i=0}^{m} F_{\text{in}}\left(\lfloor P_i' \rfloor\right) h_i'}{\sum_{i=0}^{n} h_i + \sum_{i=0}^{m} h_i'} \qquad (1)$$

を算出する．これは流線に沿ったホワイトノイズテクスチャの線積分に相当する．ここで $h_i = \int_{S_i}^{S_i+\Delta S_i} k(\omega) d\omega$，$S_0 = 0$，$S_i = S_{i-1} + \Delta S_i$ であり，$F_{\text{in}}(x,y)$ はホワイトノイズテクスチャの画素値，P_i, P_i' は正負方向の i 番目に通過する画素，$k(\omega)$ は Box フィルタ等の窓関数である．以上の操作をすべての画素について繰り返すことで，2 次元 LIC 画像が得られる．

2.2. 位相シフトによる動画生成

LIC 法により生成された $F_{\text{out}}(x,y)$ は流線を表現しているが，流れの向きは表現していない．これは LIC 計算の線積分が流れの方向を考慮しないからである．この問題は窓関数 $k(\omega)$ にハニングリプルフィルタを用いて，周期的に位相を変化させたアニメーションを生成することで解決されている．ハニングリプル関数は

$$\frac{1+\cos(d\omega+\beta)}{2}$$

で与えられる周期関数であるが，LIC は有限長の線積分であるため，トランケーションアーチファクトを減少するハニング窓関数

$$\frac{1+\cos(c\omega)}{2} \quad \{\omega : -\pi/2c \le \omega \le \pi/2c\}$$

を掛け合わせたハニングリプルフィルタ

$$k(\omega) = \frac{1+\cos(c\omega)}{2} \times \frac{1+\cos(d\omega+\beta)}{2}$$

を使用する．ここで c, d は膨張定数，β は流れを模擬するための位相シフト（ラジアン）である．計算を容易にするために，定数 a から定数 b までの積分をあらかじめ計算すると，

$$\begin{aligned}\int_a^b k(\omega) d\omega = \frac{1}{4}\Bigg[& b-a + \frac{\sin(bc)-\sin(ac)}{c} \\ & + \frac{\sin(bd+\beta)-\sin(ad+\beta)}{c} \\ & + \frac{\sin\{b(c-d)+\beta\}-\sin\{a(c-d)+\beta\}}{2(c-d)} \\ & + \frac{\sin\{b(c+d)+\beta\}-\sin\{a(c+d)+\beta\}}{2(c+d)}\Bigg]\end{aligned}$$

となる．この積分値が式(1)の重み h_i, h_i' に適用される．

2.3. LIC の 3 次元化とその問題

3D LIC ボリュームは，先に述べた 2 次元の原理をそのまま 3 次元に拡張して生成する．セル $P(x,y,z)$ における流線は，図 2 に示すように 3 次元の流線として生成され，この流線にそって 3 次元ホワイトノイズテクスチャのボクセル値をぼかし，2 次元の場合と同様のパラメタを用いて出力セル値 $F_{\text{out}}(x,y,z)$ を決定する．しかし 3D LIC の可視化法として単純に $F_{\text{out}}(x,y,z)$ のボリュームレンダリングを行う

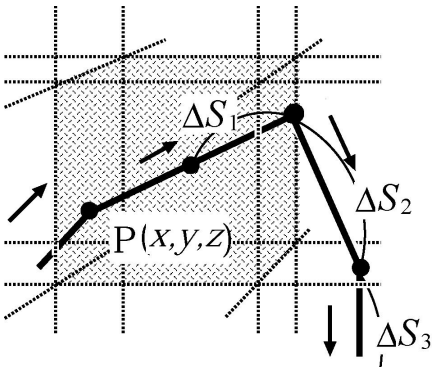

図 2 3 次元局所流線の定義

ことは, 多数の流線が視線方向に重なってしまうため, 従来は効果的でないと考えられてきた. このため, ボリュームレンダリングを部分領域に用いた手法がいくつか成果をあげているが, 一枚の結果画像からボリューム全体を理解することは困難である[2,3]. ボリュームレンダリングのスピードが格段に向上すれば, 位相シフトにより $F_{\mathrm{out}}(x,y,z)$ の時系列を生成し，周期的にボリュームレンダリングを行いながら, 対話的に視点位置や伝達関数を変更することで, 3 次元の流れが十分に把握できるのではないかと考えられる. 以下では我々が開発中の高並列計算可視化システムボリュームグラフィックス（VG）クラスタを用いて，それを検証する.

3. 3DLIC の並列化

3.1. VG クラスタ

大規模ボリュームデータを対象とした高速な計算・可視化処理を実現するために, 我々はボリュームレンダリング機能を強化した PC クラスタシステム「ボリュームグラフィックス（VG）クラスタ」を開発している（図 3）. このシステムは PC 用グラフィックスエンジンを複数使用したソートラスト型並列ボリュームレンダリングを得意とする PC クラスタである[4]. ソートラスト型に付随する通信ボトルネックを解消するため, 特別に開発したフレーム

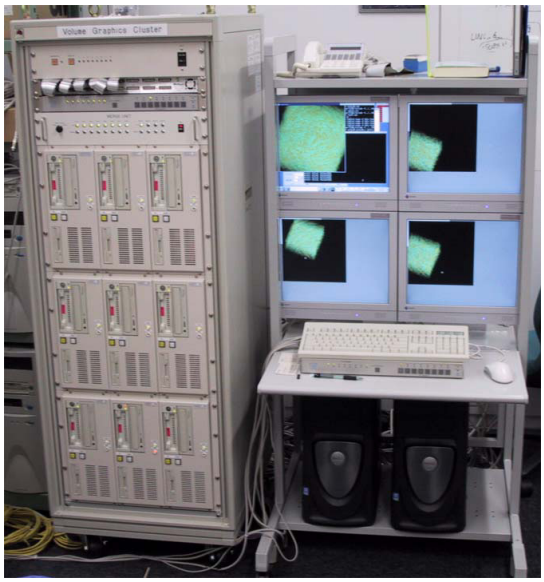

図 3 LIC アニメーション生成中の VG クラスタ

重畳装置（Image Compositing Hardware）を備えている点に特徴がある. 図 4 はホスト PC 1 台，スレーブ PC 8 台の，合計 9 台の PC から成る VG クラスタの構成例である.

まず可視化の対象となるボリュームデータが 2 進分割により空間的に均等な大きさの 8 つのサブボリュームに分解され，各スレーブ PC に分配される. 各スレーブ PC はボリュームグラフィックスエンジン（VGB）を用いてサブボリュームのボリュームレンダリング（サブイメージ）を生成する. 次に各サブイメージがスレーブ PC の PCI バス（PCI32）に挿入された専用インタフェースカード（IFB）を経由して, フレーム重畳装置に送られ, そこで先の 2 進分割情報と視点位置によって定まる優先順位にしたがった色と不透明度を考慮した合成処理（重畳処理）が行われる. 生成された画像は, ホスト PC の PCI バスのインタフェースカード（IFB）を経由してグラフィックスボード（GB）のフレームメモリに書き込まれ，画面に表示される. VGB によるサブイメージ生成処理とフレーム重畳装置へのサブイメージ転送処理は並列に実行することが可能であり,

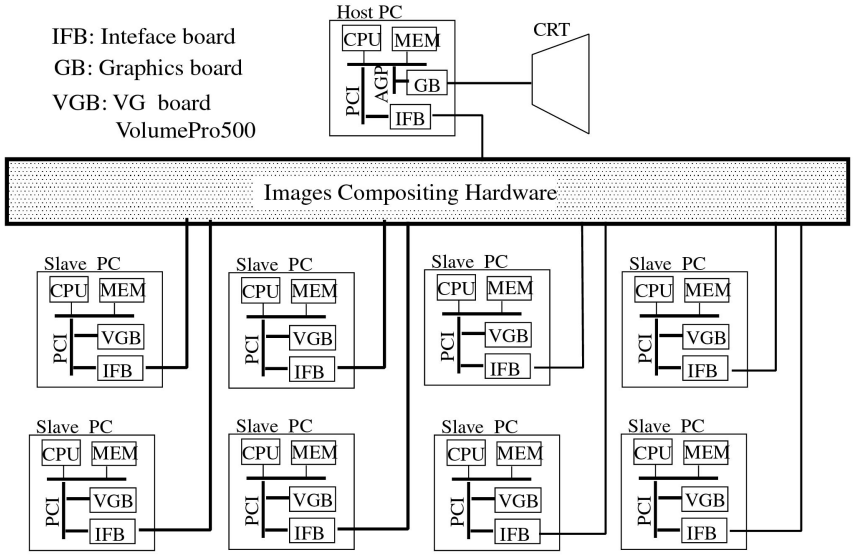

図 4　VG クラスタ（9PC システム）の構成図

ダブルバッファリングによりサブイメージ転送時間をサブイメージ生成時間の中に隠蔽することができる. これによりアニメーション生成時にサブイメージ転送時間が無視できるので，VG クラスタは VGB 単体のレンダリング速度で動画像を生成できる. フレーム重畳装置単体では最大 8 つまでの PC のサブイメージしか重畳できないが, オクトリー状に接続したフレーム重畳装置によって，原理的には VGB の基本描画性能を低下させることなくいくらでも多くの PC を使った並列ボリュームレンダリングシステムが実現できる.

我々のプロトタイプシステムで使用している VGB （TeraRecon, Inc. 製 VolumePro 500）は最大 256^3 ボクセルのボリュームデータを 512×512 の画面サイズで 30 フレーム／秒以上の速さで描画できる他, ボリュームメモリを分割して複数のボリュームデータを格納し連続に描画するアニメーション生成機能を持つ. したがってフレーム重畳装置の階層接続によって, 任意サイズの静止ボリュームデータや動ボリュームデータをリアルタイムで可視化することが可能である.

3.2.　3D LIC 計算と可視化の並列化

VG クラスタはボリュームレンダリングを行わない限り単純な分散メモリ型並列計算機としても使えるので，$F_{\mathrm{out}}(x,y,z)$ を生成する処理の並列化には二通りの方法が考えられる. 一つはボリュームレンダリングと同様に空間を分割する方法, もう一つはアニメーション生成時の位相シフト値毎に並列化する方法である. 前者はレンダリングと整合性が良く，3 次元ベクトル場の生成と同時に可視化を行うシミュレーションステアリング等に都合がよい. しかし 3D LIC 生成は時間のかかる処理であり，現状の PC の計算性能では VolumePro 500 のフレームレートに相当する速さで 3 次元ベクトル場から 3D LIC ボリュームを生成することは難しい．また 3D LIC では約 l ボクセルにわたって流線を追跡する必要があるため, 空間分割によって流線が切断されることがないように, $F_{\mathrm{out}}(x,y,z)$ の計算のためのサブボリュームは, レンダリングに必要なサイズのサブボリュームよりも周囲に l ボクセル程度大きめに取る必要がある. さらに, 分割生成された $F_{\mathrm{out}}(x,y,z)$ をハードディスク等に保存するには, 各スレー

表 1 3D LIC 時系列データ

	Data1	Data2	Data3
Num. of voxels	135^3	180^3	240^3
Num. of phases	32	16	8
Num. of vortices	2	3	4
l	60	40	20
Granularity	2	2	4

表 2 アニメーション生成速度

		512^2	768^2
Data1	Rend. Time [*ms*]	19.4	34.8
	Merge Time [*ms*]	9.3	20.9
	Frame Rate	51.5	28.7
Data2	Rend. Time [*ms*]	19.4	34.8
	Merge Time [*ms*]	9.3	20.9
	Frame Rate	51.5	28.7
Data3	Rend. Time [*ms*]	19.3	34.7
	Merge Time [*ms*]	9.3	20.9
	Frame Rate	51.8	28.8

ブ PC 上のデータを一箇所に集め，並べ替える作業が必要になる．

位相シフト値毎に並列化する方法は空間分割の必要がない半面，一つの PC でボリュームデータ全体を処理する必要があり, 大規模なデータを扱う場合にはメモリの制限を受けやすい．

そこで本稿では, 一つの PC で $F_{\mathrm{out}}(x, y, z)$ が計算でき，単体の VolumePro 500 では可視化できない程度の大きさの 3D LIC 時系列を位相シフト値毎に並列生成し，VG クラスタプロトタイプを用いて空間分割による対話的なアニメーション生成を行うことにする．

4. 実験

渦のシミュレーションプログラムを用いて，表 1 に示す 3 種類の 3 次元ベクトルデータを生成した．Num. of vortices はデータ中の渦の数，l は LIC 計算時の流線長の閾値，Granularity はホワイトノイズテクスチャのボクセルサイズである．ボクセル数（Num. of voxels）と位相シフト数（Num. of phases）は，3D LIC 時系列の大きさを示す．つまり Data1 は 135^3 ボクセルのボリュームデータ 32 個からなる時系列データである．これらの値は，各スレーブ PC の VolumePro 500 のメモリをできるだけ有効に使えるように設定した．各 PC で異なる位相シフト値で計算された 3D LIC ボリュームは，NFS で共有されたホスト PC のハードディスクに並列に書き込まれ時系列データとなる. Data1, 2, 3 の 3D LIC 時系列の生成時間は CPU（Pentium III 933MHz, メモリ 512 MB）を 8 個使用してそれぞれ 83 分，60 分，36 分であった．

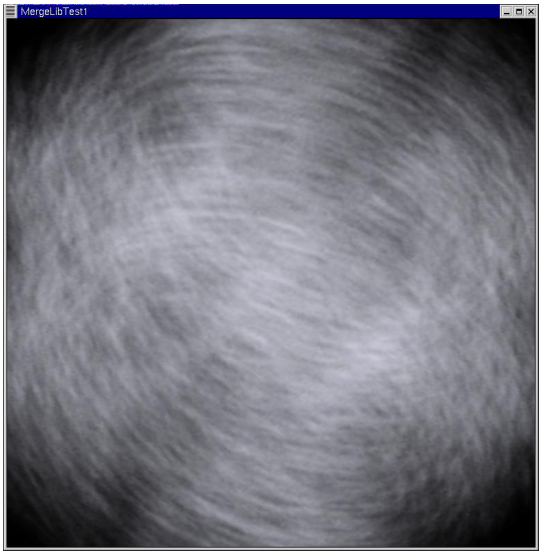

図 5 VG クラスタによる 3D LIC ボリュームレンダリングの 1 コマ（Data2 768^2）

次にホスト PC 上の 3D LIC 時系列データを VG クラスタの空間分割機能により 512^2, 768^2 の 2 種類の画面サイズで可視化した結果が表 2 である．ここで Rend. Time はサブイメージ生成時間，Merge Time はフレーム重畳装置によるサブイメージ重畳時間であり, 後者はパイプライン処理により隠蔽されるため, アニメーション生成時のフレームレートは Rend. Time の逆数になる. 画面が大きいほど描画速度は遅くなったが, ボクセル数は描画速度にほとんど影響しなかった．図 5 は Data2 を 768^2 の画面サイズで描画したアニメーションの 1 コマであ

る．静止画ではわからないが，アニメーションで繰り返し描画し，対話的に視点位置を操作することにより，渦の 3 次元構造が明瞭に確認できた．また，l が長いほど 3D LIC が長い尾を引くため，渦同士の関係が明瞭になることや，位相シフト数が少ない程 1 周期のフレーム数が少なくなりダイナミックに見えることなども確認できた．これらのことは他のデータ，画面サイズでも同様に確認された．

本実験で用いたソフトウェアは，TeraRecon, Inc. 製 VolumePro500 用ライブラリ（VLI 1.1），三菱プレシジョン㈱が開発中の VG クラスタ用フレーム重畳装置 API（MergeLib）を用いて PC クラスタコンソーシアム製 SCore 3.3 (RedHat Linux 6.2J ベース)上で C++で記述した．

5. まとめ

LIC 法を 3 次元化し VG クラスタに実装することにより，大規模な 3 次元ベクトル場データから 3D LIC 時系列を並列に生成し，ボリュームレンダリングのアニメーションを対話的に操作できることを確認した．これによって，従来効果的でないとされてきた単純な 3D LIC のボリュームレンダリングが，視点位置や伝達関数を対話的に変更できる環境においては，高解像度 3 次元流の非常に有効な可視化方法となり得ることが示された．今後さらに，流速表示技法[5]，流線照明モデル[6]，選択的ボリュームレンダリング[7,8]等の技術を取り入れることにより，より効果的に 3 次元流を可視化できるようになると考えられる．

今回生成した 3D LIC 時系列は単一の VolumePro 500 ではリアルタイムレンダリングが不可能な大きさであったが，VG クラスタシステムの高いスケーラビリティーにより，VolumePro 500 のリアルタイムレンダリング性能を保ったままメモリ制限を克服できることが示された．VG クラスタはパイプライン処理により，重畳処理時間をサブイメージレンダリング時間に隠蔽し，アニメーション時のフレームレートを高くできるので，本研究のようなアニメーション生成が必須のアプリケーションには特に有効である．今後はより複雑な流れデータや，非定常流などの計算と可視化に VG クラスタシステムを応用して行きたい．

謝辞

本研究は科学技術振興事業団計算科学技術活用型特定研究開発推進事業（13-D4）の成果である．

参考文献

[1] Cabral, B., Leedom, L.: Imaging vector field using line integral convolution, *Computer Graphics (Proc. SIGGRAPH 93)*, August 1993, pp.263-270.

[2] Stalling, D., Zockler, M., Hege, M.: Parallel line integral convolution, *Parallel Computing*, vol.23, no.7, 1997, pp.975-989.

[3] Mao, X., Kikukawa, M., Fujita, N., Imamiya, A.: Line integral convolution for 3D surfaces, *Visualization in Scientific Computing '97*, France, 1997, pp.57-69, *Proc. Eurographics Workshop in Boulogne-sur-Mer*, Springer-Verlag.

[4] Muraki, S., Ogata, M., Ma, K-L., Koshizuka, K., Kajihara, K., Liu, X., Nagano, Y., Shimokawa, K.: Next-generation visual supercomputing using PC clusters with volume graphics hardware devices, *CD-ROM Proc. IEEE SC2001*, November 2001.

[5] 鈴木靖子，藤代一成，竹島由里子: LIC テクスチャマッピングを用いた 3 次元流れ場の対話的可視化 3，第 60 回情報処理学会全国大会，拓殖大学八王子キャンパス，March 2000, 3ZA-06.

[6] Zockler, M., Stalling, D., Hege, M.: Interactive visualization of 3D-vector fields using illuminated stream lines, *Proc. IEEE Visualization '96*, San Francisco, October 1996, pp.107-114.

[7] Suzuki, Y., Fujishiro, I., Chen, L., and Nakamura, H.: Hardware-accelerated selective volume rendering of 3D LIC textures, To appear *Proc. IEEE Visualization 2002*, Boston, October - November 2002.

[8] Chen, L., Fujishiro, I., Suzuki, Y.: Comprehensible volume LIC rendering based on 3D significance map, *Proc. SPIE Conference on Visualization and Data Analysis 2002*, San Jose, January 2002, pp.142-153.