

2010年度
卒業論文

CAVE型VR装置を用いたベクトル場の
対話的可視化手法

神戸大学工学部情報知能工学科
吉崎健一

指導教員 陰山聰

2011年2月23日

CAVE型VR装置を用いたベクトル場の対話的可視化手法

吉崎健一

要旨

ベクトル場を解析するための可視化手法、『没入的力線表示法』を開発した。これはCAVEと呼ばれる方式の没入型バーチャルリアリティ(VR)装置の内部で、対話的かつリアルタイムに複数の力線を表示させる手法である。力線を追跡する出発点(初期条件)は、CAVEのVR空間中に立つ解析者が、手に持ったコントローラ(3Dマウス)を3次元的に動かすことで自由に指定することができる。それらの出発点を通る力線群を可視化するために、正負両方向に力線方程式を数値的に積分する。出発点の指定と数値積分は力線の3次元表示と同時にリアルタイムで計算されるので、解析者がコントローラを任意の方向に動かすと、それに応じて力線群があたかも生き物のようにダイナミックに動く様子を観察することができる。各力線には色のパターンが埋め込まれており、そのパターンの移動方向によって力線が示すべきトル場の向きを表す。

目 次

1 序章	1
2 バーチャルリアリティシステム CAVE	2
2.1 バーチャルリアリティ	2
2.2 CAVE	2
2.3 可視化プログラムの基本ツール	2
2.4 pCAVE	3
3 基本手法	8
3.1 力線の計算法	8
3.2 4次精度ルンゲ・クッタ・ギル法	8
4 没入的力線表示法	10
4.1 目的	10
4.2 特色	10
4.3 力線出発点の決定法	11
4.4 力線の時間積分法	11
4.5 力線の描画	12
4.6 色分布パターンの移動による流れ方向の提示	15
4.7 地球ダイナモへの応用例	16
5 まとめ	20
6 謝辞	20

1 序章

計算機シミュレーションの出力は膨大な量の数値データである。この中から情報を引き出し、計算機の中でどのような構造・現象が再現されたかを知るために、出力数値データを画像データに変換する、可視化と呼ばれるプロセスが不可欠である。

シミュレーションデータの可視化には様々な種類があり、データの種類や確かめたい構造・現象によって用いる可視化手法が異なる。シミュレーションデータには大きく分けて二つの種類がある。すなわちスカラー場とベクトル場である(2階以上のテンソル場が可視化の対象となることは今のところまれである)。スカラー場とは、離散化された空間に、方向などの情報を持たない数値が与えられた場である。例えば液体中の温度分布のシミュレーションデータでは、場の各点にその位置での温度というスカラー値が与えられている。このスカラー場を可視化する手法としては、等値面やボリュームレンダリング、断面図等が挙げられる。

一方ベクトル場とは、離散化された空間の各点において、データがベクトル値で与えられた場である。例えば液体中の熱対流のシミュレーションデータでは、場の各点にその位置での流れの速度が、それぞれベクトルとして与えられている。ベクトル場を可視化する手法としては、力線、パーティクルトレーサー、Line Integral Convolution 法 (LIC 法) 等が挙げられる [1][2][3]。

このうち力線は、シミュレーションデータが例えば磁場であれば磁力線を表す可視化手法であり、ベクトル場のある部分から連続する流れの形状を直感的に知る手法として優れている。しかし一方で、まず流れの出発点をあらかじめ定めておかなければ、そこから導出される力線を得ることができないという欠点を持つ。例えば広大な3次元ベクトル場の、どこかに特徴的な流れがあり、それを探し出したいと考えたとする。この時、考えられる方法の一つとしては、網羅的に力線の出発点を与えてその特徴的な流れを求める方法がある。しかしこの方法は、計算量の上でも、またそれにより得られる無数の不必要的な力線から目当ての力線を探し出すという労力の上でも、さらに網羅的に与える力線出発点を微妙にずらしたいという要求が生じる場合を考えても、現実的ではない。加えて、複雑な3次元ベクトル場から得られる力線の構造を完全に把握し、その特徴を目視するためには、力線を2次元的な方法で表示したのでは不十分である。この問題を克服することが本研究の目的である。

本研究ではバーチャルリアリティ(VR)技術を活用した力線の新しい可視化手法を提案する。VR技術とは、立体画像を用いて、数値データからコンピュータグラフィックスを用いて作られた仮想現実世界を、実在するかのように提示し、体験者にその仮想現実世界への没入感を与える技術である。この技術により、可視化されたシミュレーション結果を広い視野角で立体視し、リアルタイムでの反応

が得られる対話的解析が可能となる。VRについては次章で説明する。

2 バーチャルリアリティシステム CAVE

2.1 バーチャルリアリティ

計算機シミュレーションの結果生成される数値データの3次元的な可視化のためには、コンピュータグラフィックスの技術を用いて一般的な3次元グラフィックスを作成するだけでは不十分である。3次元グラフィックスを2次元的な画面で解析したのでは、データに潜む複雑な3次元構造や時間発展を完全に把握することはできないからである。

理想的な可視化手法の一つの候補がバーチャルリアリティ(Virtual Reality, VR)技術を用いた手法である。これはシミュレーション結果を、コンピュータグラフィックスを用いて作られた仮想現実世界に立体的に提示する技術である。体験者が仮想現実世界を実在しているように感じる感覚、没入感を得るためにには、広い視野角での立体視や、仮想現実空間の物体を操作できる対話性、そして画像が体験者の行動に応じてリアルタイムに反応することが重要である。

2.2 CAVE

イリノイ大学シカゴ校において開発されたVR装置CAVEは、スクリーン投影方式のVRシステムのパイオニアであり、様々なVR方式の中でも特に高い没入感を得ることができる。スタンダードなCAVEシステムの中心部は、正面、左右の壁面、及び床面の全てが一辺10フィートのスクリーンとなっている立方体の部屋である。各スクリーンにステレオプロジェクタで立体映像を投影する。体験者は立体眼鏡をつけこの部屋の中に立つ。専用のコントローラはワンドと呼ばれ、体験者が手を持って使用する。CAVEにはトラッキングシステムが組み込まれており、立体眼鏡およびワンドの位置と方向をリアルタイムで検知することができる。そのトラッキングシステムで得られた情報をもとに、スクリーンに投影される画像がリアルタイムに調整される。たとえば体験者がしゃがんだり見下ろしたりしても、体験者の視点に応じてそこから見えるべき画像がリアルタイムに表示される。

2.3 可視化プログラムの基本ツール

本研究では、前節で述べたようなトラッキングシステム等のCAVEハードウェアのアプリケーションプログラムインターフェース(API)として、CAVEライブ

ラリ (CAVE Lib) を利用した。CAVE Lib はスクリーンどうしの映像をなめらかに繋げることや、体験者の視点位置に応じた適切な射影変換の実行など、CAVE 装置の VR 空間構成に必須の処理する。

またコンピュータグラフィックスの API としては Open Graphics Library (OpenGL) を利用した。OpenGL は、現在のところコンピュータグラフィックスの事実上の標準であり、各種のハードウェアプラットフォームに実装されている [4]。OpenGL では仮想的な 3 次元空間中に物体の頂点となる座標を与え、その座標を線や面で結ぶことによって仮想的な 3 次元オブジェクトを構築する。作られたオブジェクトには陰影処理などに用いる法線ベクトルや、質感を決める反射率等のさまざまな光学特性を与えることができる。そして光源の位置や方向そしてその光の色、視点の位置と方向を指定することによって、射影変換された画像が自動的に計算される。OpenGL にはウインドウマネジメントやマウスイベント等ユーザー入力を取得するコマンドは含まれていない。そのため得られた画像を表示するためには、OS やウインドウシステムに依存する何らかの手段が別に必要となる。

CAVE Lib には前述した通り、視点の位置と方向の取得、そこから見えるべき画像の描画を行う機能が含まれているため、これらを用いることによって、そうした設定をプログラマが意識することなく、OpenGL で構築した物体を自動的にスクリーンに描画することができる。

さらに本研究ではVFIVE[5][6][7] と呼ばれる CAVE 用の VR 可視化フレームワークを利用する。VFIVE は本研究の指導教員である陰山教授が開発した CAVE 用 VR 可視化ソフトウェアである。VFIVE には、“Particle Tracer”、“Local Arrows”、“Spotlight Snow” 等のベクトル場用可視化機能が既に実装されている。本研究で開発する新しいベクトル場可視化機能も、VFIVE のメニューに登録し、その可視化機能の一つとして組み込んだ。

2.4 pCAVE

本研究では神戸大学にある一面型の CAVE 装置 “pCAVE” を使用した (Fig. 1 ~ 6)。当装置のスクリーンは正面の一つのみであり、スクリーンの背後にあるステレオプロジェクターから画像を投影する。プロジェクターから投影される画像は右目用画像と左目用画像が高速で交互に映し出されており、立体眼鏡はその切り替えに同期して左目右目にシャッターを下ろし、体験者の右目には右目用画像だけ、左目には左目用画像だけを見せる。この操作が高速であるため、体験者の目には右目用画像と左目用画像がそれぞれの目に同時に映し出されるように見え、そのため画像を立体画像として認識することができる。

立体眼鏡とワンドの位置情報の取得には超音波センサが用いられ、傾き情報の取得にはジャイロが使われている。立体眼鏡とワンドに組み込まれた超音波セン

サが、スクリーン上部および左右に取り付けられた超音波発生装置から発せられる超音波を受信することで、スクリーンに対する現在の位置の情報をそれぞれ取得できる。また傾き情報の取得に用いられるジャイロも、同様に立体眼鏡とワンド内部に組み込まれている。これら視点およびワンドの位置や方向、傾きの情報は、位置センサー処理用 PC によって常に解析され、画像処理用のグラフィックワークステーション (GWS) にその解析結果が送られる。GWS ではその情報をもとに右目用と左目用の画像を作り、それをプロジェクターに送る。この一連の処理が常に行われることによって、体験者の視点移動に応じたリアルタイムな画像変換が実現される。またワンドには 5 つのボタンとジョイスティックがついており、それらの操作情報を取得される。ワンドの各ボタンについて、押された、押されているといった複数の状態の情報をそれぞれ取得できる。

pCAVE の構成を Table1 にまとめる。スクリーンは半透明 (リアプロジェクション) 型で、そのサイズは 10.0feet x 7.5feet である。プロジェクターはクリスティ・デジタル・システムズ社製の「Mirage S+4K」1 台、超音波位置センサーおよびワンドは Intersence 社製の「IS900」、GWS は日本 SGI 社製の「Asterism Deskside ADT08C」、立体視のための液晶シャッター眼鏡から構成されている。

Projector	Mirage S+4K
Tracking System	IS900
GWS	
Asterism Deskside ADT08C	
CPU	AMD Opteron 2350 x2
Memory	64GB
GPU	nVIDIA Quadro FX 4600

Table 1: Specification of pCAVE System



Fig. 1: Screen for pCAVE



Fig. 2: Stereo projector of pCAVE System



Fig. 3: pCAVE's Graphics workstation for the rendering



Fig. 4: PC for tracking system of pCAVE



Fig. 5: Liquid shutter glasses with the tracking sensor



Fig. 6: Portable controller “ wand ” for pCAVE

3 基本手法

本章では、本研究で開発した没入的力線表示法の基礎となる2つの手法について解説する。

3.1 力線の計算法

力線とは、各瞬間ににおけるベクトル場に対して、各点が常にその位置におけるベクトルに平行になるように引かれた曲線である[8]。つまり、ベクトル場 \mathbf{a} の力線は、パラメータ s を用いて

$$\frac{d\mathbf{x}(s)}{ds} = \mathbf{a}(\mathbf{x}(s))$$

で定義される曲線 $\mathbf{x}(s)$ である。特に \mathbf{a} が磁場の場合は磁力線と呼ばれる。上式は微少量 Δs を使えば、 Δs の1次のオーダーで近似的に以下のように書くことができる。

$$\begin{aligned}\Delta x &= a_x(\mathbf{x}(s))\Delta s \\ \Delta y &= a_y(\mathbf{x}(s))\Delta s \\ \Delta z &= a_z(\mathbf{x}(s))\Delta s\end{aligned}$$

初期点における \mathbf{a} の x 、 y 、 z 成分を求め、それを Δs で積分し続けることによって、ベクトル場の方向に Δs ずつ進んだ点が次々得られる。こうして得られた点列をつなぎだ線が力線である。

ここでは力線の定義式である微分方程式を数値的に解く方法を、1次オイラー法による数値積分法に基づいて説明した。しかしながらこの方法は $O[(\Delta s)]$ の精度しかない。本研究で開発した没入的力線表示法では、精度を上げるために4次精度のルンゲ・クッタ・ギル法で数値積分を行っている。

3.2 4次精度ルンゲ・クッタ・ギル法

本研究では力線の積分に4次精度のルンゲ・クッタ・ギル法を用いた。ルンゲ・クッタ・ギル法とは、丸め誤差の累積を少なくするように任意定数を選ぶルンゲ・クッタ法の一種であり、コンピュータのメモリの使用量が節約できるという利点がある[9]。

ルンゲ・クッタ・ギル法の公式と計算手順について説明する。簡単のため

$$\frac{dy(x)}{dx} = f(x, y(x)) \quad (y(x_0) = y_0)$$

という常微分方程式を、きざみ幅 h で積分することを考える。このとき 4 次ルンゲ・クッタ・ギル法では差分式は以下のように決まる。

$$\begin{aligned} y_{k+1} &= y_k + \frac{1}{6}(k_1 + (2 - \sqrt{2})k_2 + (2 + \sqrt{2})k_3 + k_4) \\ k_1 &= hf(x_k, y_k) \\ k_2 &= hf\left(x_k + \frac{h}{2}, y_k + \frac{k_1}{2}\right) \\ k_3 &= hf\left(x_k + \frac{h}{2}, y_k + \frac{\sqrt{2}-1}{2}k_1 + \frac{2-\sqrt{2}}{2}k_2\right) \\ k_4 &= hf\left(x_k + h, y_k - \frac{\sqrt{2}}{2}k_2 + \frac{2+\sqrt{2}}{2}k_3\right) \end{aligned}$$

上式を用い、小さい記憶容量で y_{k+1} を求めるアルゴリズムは以下のように書ける。
まず次式を順次計算し

$$\begin{aligned} \bar{k} &= k_1 \\ \bar{y} &= y_k + c01 * \bar{k} + c02 * \bar{q} \\ \bar{q} &= c03 * \bar{q} + c04 * \bar{k} \end{aligned}$$

同様の手順で、 \bar{k} 、 \bar{y} 、 \bar{q} の値を更新する。

$$\begin{aligned} \bar{k} &= k_2 \\ \bar{y} &= \bar{y} + c05 * \bar{k} + c06 * \bar{q} \\ \bar{q} &= c07 * \bar{q} + c08 * \bar{k} \end{aligned}$$

$$\begin{aligned} \bar{k} &= k_3 \\ \bar{y} &= \bar{y} + c09 * \bar{k} + c10 * \bar{q} \\ \bar{q} &= c11 * \bar{q} + c12 * \bar{k} \end{aligned}$$

$$\begin{aligned} \bar{k} &= k_4 \\ \bar{y} &= \bar{y} + c13 * \bar{k} + c14 * \bar{q} \\ (\bar{q} &= c15 * \bar{q} + c16 * \bar{k}) \end{aligned}$$

\bar{q} の初期値は 0 として与えておく。最後の \bar{q} は求める必要の無い項であるが、ループプログラムによって最初の段の計算式を繰り返し利用する場合には、c15 および c16 の係数を与えておく。

また ($c01, c02, c03, \dots, c16$) の各係数は積分計算ルーチン以前に求めておき、定数として与える。この操作により積分ルーチンの計算量が減少する。なお各係数の値は以下の通り。

$$\begin{array}{llll} c01 = \frac{1}{2} & c05 = \frac{2 - \sqrt{2}}{2} & c09 = \frac{2 + \sqrt{2}}{2} & c13 = \frac{1}{6} \\ c02 = 0 & c06 = -\frac{2 - \sqrt{2}}{2} & c10 = -\frac{2 + \sqrt{2}}{2} & c14 = -\frac{1}{3} \\ c03 = 0 & c07 = \frac{3\sqrt{2} - 4}{2} & c11 = -\frac{3\sqrt{2} + 4}{2} & c15 = 0 \\ c04 = 1 & c08 = 2 - \sqrt{2} & c12 = 2 + \sqrt{2} & c16 = 0 \end{array}$$

最後に得られた \bar{y} が、求める y_{k+1} である。記憶用変数として用意しなければならないのは \bar{k} 、 \bar{y} 、 \bar{q} の三つで良い。通常ルンゲ・クッタ法では各段の \bar{k} それぞれに記憶領域を確保しなくてはならないが、ルンゲ・クッタ・ギル法はルンゲ・クッタ法に比べて少ない記憶領域で計算できるという利点があり、計算機を用いた数値計算に適している。

4 没入的力線表示法

本章では、本研究で開発した没入的力線表示法について説明する。

4.1 目的

複雑な 3 次元ベクトル場の中から興味深い空間構造を効率的に見出し、直観的に可視化することが本手法の目的である。ベクトル場の基本的可視化手法の一つである力線を基礎とし、これを発展させて、3 次元力線を解析できるようにする。また、ベクトル場中の任意の場所に、対話的に力線を作れるようにすることで解析の効率を上げる。それらの実現のためにバーチャルリアリティ技術を利用する。

4.2 特色

CAVE 型 VR 装置のリアルタイムな反応と対話性を利用し、任意の場所で、瞬間的かつ連続的にベクトル場の可視化を行うことが本手法の特徴である。本手法

を実現するプログラムは CAVE Lib 及び OpenGL を用いており、さらに VFIVE のフレームワークに組み込むことで CAVE 装置に実装している。

利用者はワンドを使って、スクリーン上の任意の場所に流れを横切る線を作り、その線上から一斉に複数の力線を描く。各力線はある一定時間、時間積分した地点までの座標情報にもとづいて描画される。力線の出発点はワンドの位置と向きに応じてリアルタイムに変化し、それに伴って描画される力線もリアルタイムに更新される。

4.3 力線出発点の決定法

以下でプログラム内部での実装方法について説明する。まず、ワンドの先端からワンドの向いている方向へとまっすぐ伸びる仮想的な線を作り、その上に等間隔で力線の出発点を並べる。各出発点は、正の方向へ積分される力線と、負の方向へ積分される力線の二つの出発点が重なっており、利用者には一つの出発点として見える。ワンドの向きや座標が変化すると、仮想的な線とその上の各出発点もリアルタイムに追従する。ワンドの向きと座標は CAVE 型 VR 装置のトラッキングシステム及び CAVE Lib によって自動的に検出される (Fig.7)。

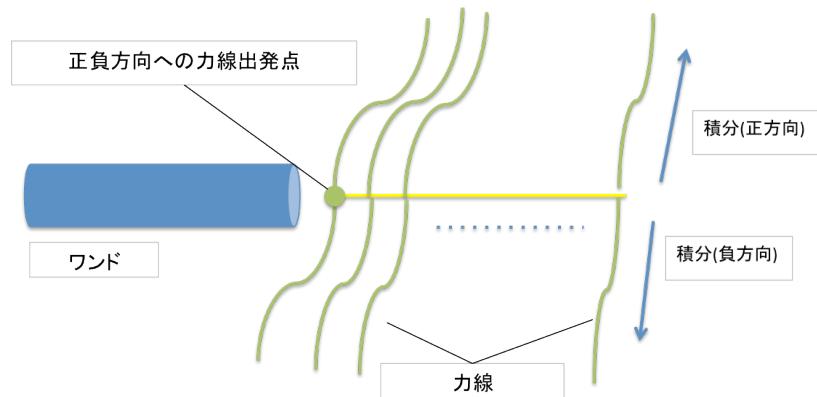


Fig. 7: Seeding method of the lines of force. Seeding points are located on a straight line emitted from the wand controller.

4.4 力線の時間積分法

次に、速度ベクトル場の数値積分による力線を表す座標列の取得について述べる。各力線はその軌跡となる点列を記憶しておく配列を持つ。まず、最初の出発

点の座標を配列に記憶する。続いて最初の出発点から、あるきざみ幅 h で積分された座標を求め、その座標を次の出発点とする。この操作を 5 回繰り返すごとに、5 回目に得られた座標を順次配列に記憶してゆく (Fig.8)。5 回の積分おきに座標を記録する理由は、後述する力線の描画の計算において、1 回の積分間隔で力線を表す曲線を描画することに比べ、少ない計算量で描画できるようにするためにある。きざみ幅が微小なとき、1 回の積分間隔で繋げられた曲線を描画した場合と 5 回の積分間隔で繋げられた曲線を描画した場合とでは、VR 装置上での見え方に大きな差異は無い。力線の点列を記録する配列の大きさは、各力線に共通の大きさを与え、その配列が一杯になるまで座標を取得したところで積分を終える。

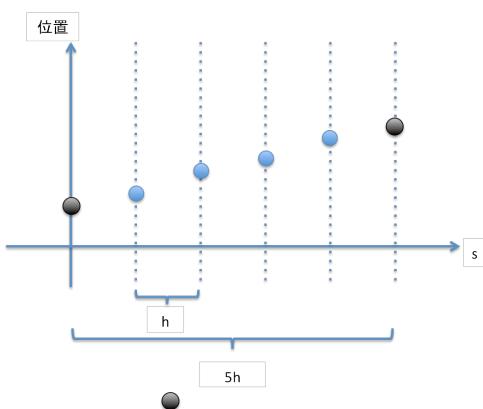


Fig. 8: Numerical integration of the line of force. Position data are stored for every five steps.

各座標の速度はユーザーが指定した入力流れ場のデータから、空間 3 次精度の補間によって求める。速度の時間積分には 4 次精度のルンゲ・クッタ・ギル法を用いる。入力データの読み込み、空間補間および時間積分の適切なきざみ幅の導出はすべて VFIVE が処理する。

4.5 力線の描画

時間積分によって得られた座標データ列から、一本ずつ力線を描画する。一本の力線の隣り合う座標同士を円柱で結び、それらを繋げてチューブ状の曲線を作ることによって力線を表現する。曲線を円柱でチューブ状に描く理由は、VR 装置上での視認性を高めるためである。

ここで OpenGL による円柱の描画について説明する。OpenGL では各頂点ごとに法線ベクトルを指定することができ、さらに、その各頂点が作る面上の法線が

各頂点の法線から自動的に補間されるという機能がある。物体の陰影処理は法線から計算されるため、この機能を用いることによって、多角柱を円柱のように描画することができる。よって OpenGL で円柱を描くには、側面が曲面になるよう各頂点に適切な法線を与えた多角柱を作成すればよい。

次にその適切な法線の求め方を説明する。今、3次元空間中の隣り合う2点 P_1 と P_2 があり、その間に円柱を描きたい。このとき、まず P_1 から P_2 に向かう3次元のベクトル T を求める。次に T とは異なる任意のベクトル W を用意し、 W と T の外積を求め、求めた外積の方向へ向かう単位ベクトルを U とする。続いて T と U の外積も求め、求めた外積の方向へ向かう単位ベクトルを R とする。すると T 、 U 、 R は互いに直交するベクトルとなり、 U および R の作る平面は T に垂直な平面となる。そこで、 P_1 および P_2 を中心とした T に垂直な半径 r の円周上の点は、次式で表される。

$$\begin{aligned} A_1(\theta) &= P_1 + r(\sin(\theta) * U + \cos(\theta) * R) \\ A_2(\theta) &= P_2 + r(\sin(\theta) * U + \cos(\theta) * R) \end{aligned}$$

この円周上に n 角の多角形の頂点を求めるには、 $\theta = 2\pi/n$ とすればよい。こうして P_1 、 P_2 を結ぶ多角柱の各頂点の座標 $A_1(\theta)$ 、 $A_2(\theta)$ が得られた。最初に求めていた、多角柱を構成する各頂点の適切な法線は $\sin\theta U + \cos\theta R$ である (Fig.9)。

法線情報を含む多角柱の頂点座標を求めた後、最後に多角柱の面を描画する。OpenGL が描くことのできる面は三角形のみであるため、多角形の面を高速に描くためには、多角形を三角形の集合で表現し描かなければならない (Fig.10)。

以上の操作を全ての点について行うことにより、チューブ状の力線を描画する。

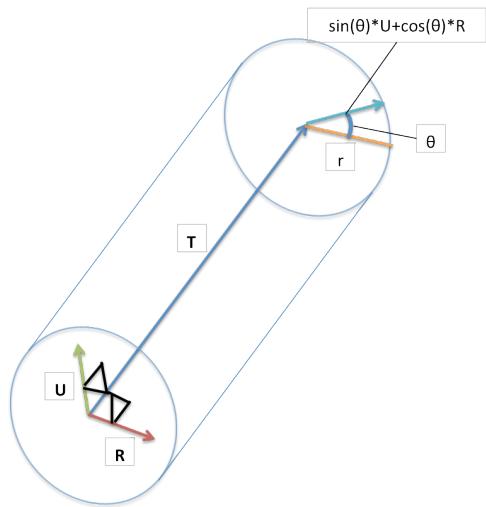


Fig. 9: Cylinder construction with OpenGL. 円柱を描くための多角柱の頂点の法線ベクトルを求める

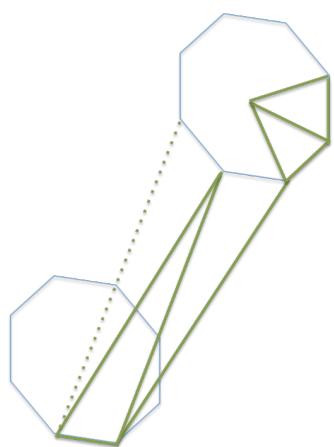


Fig. 10: Cylinder construction with OpenGL(2) . 多角柱の側面を三角形で分割しそれぞれ塗りつぶす

4.6 色分布パターンの移動による流れ方向の提示

本研究で提案する可視化手法は、ベクトル場の正の方向への積分結果と負の方向への積分結果を同時に可視化するものである。可視化された軌跡が正負どちらの向きに積分されてできたものであるかを判別できるようにした。

その実装には以下の方法を用いた。まず、チューブを構成する各円柱を描画する過程で、ある間隔で周囲と異なる色の側面を持つ円柱を描いた。この側面の特殊な色は、プログラム稼働時間に合わせて、順次流れの正の方向にある隣の円柱に引き継がれる。この操作により、流れの正方向に移動する縞模様が力線表面に描画される (Fig.11)。

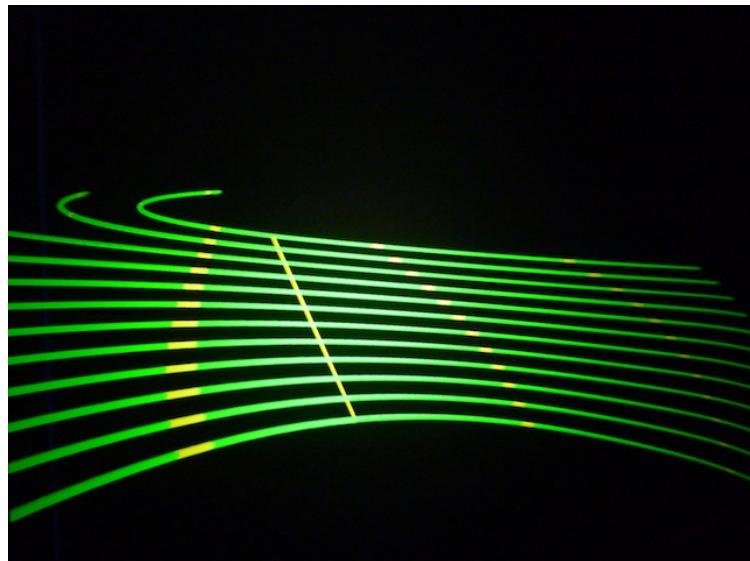


Fig. 11: Drifting color pattern on lines of force. 中央の黄色の直線が力線出発点。左右に広がる緑色の力線の中で、黄色になっている部分が流れの向きを表す縞模様である

4.7 地球ダイナモへの応用例

地球ダイナモとは、地球内部の核（コア）領域にある液体鉄が、対流運動によって大規模な磁場を作りだす現象である [10]。地球ダイナモの結果生じる地磁気逆転の物理機構は、地球科学に残された重要な未解決問題であり、スーパーコンピュータを駆使した大規模な計算機シミュレーション研究が近年盛んに行われている。この地球ダイナモの解明を目的に行われたシミュレーションの結果求められた地球の核内での流体金属の流れ場に対して、本プログラムを実行した。可視化操作中の様子と、可視化された力線の例を Fig12 ~ 17 に示す。

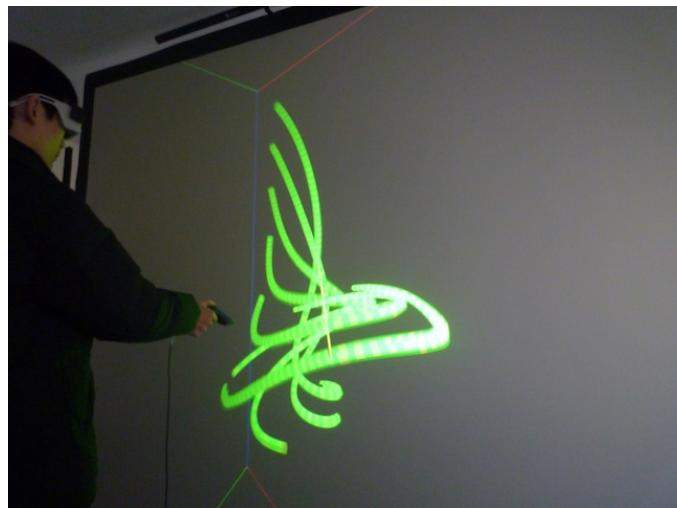


Fig. 12: A snapshot of the proposed visualization on pCAVE.

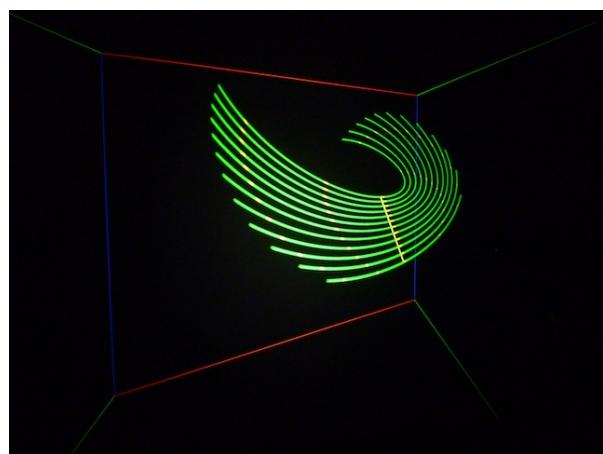


Fig. 13: A snapshot of the proposed visualization method applied to geodynamo simulation data.



Fig. 14: Another example of the proposed visualization method. 流れ場が急速な渦状になっている様子がわかる

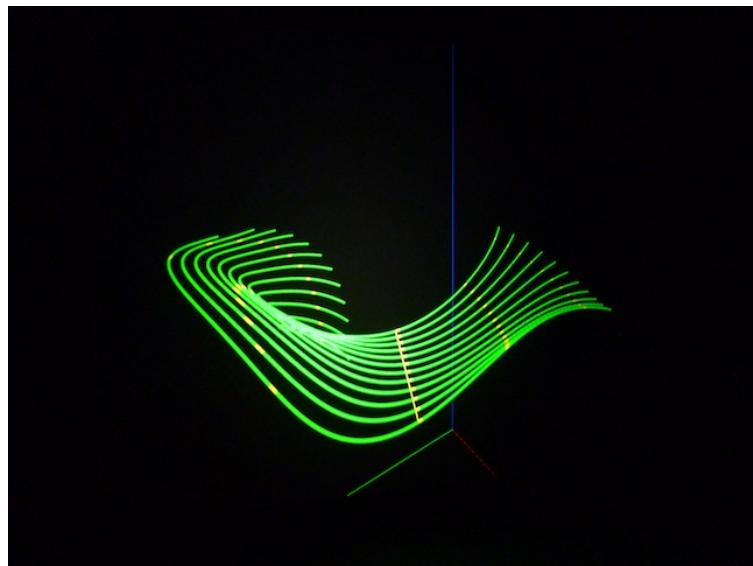


Fig. 15: Another snapshot. 複雑なうねりの様子が面となって見える

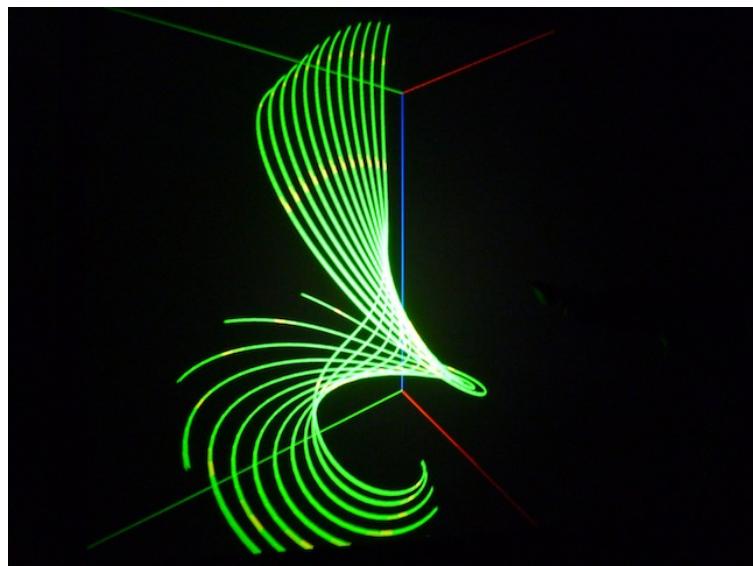


Fig. 16: Another snapshot. 複雑なねじれの構造がわかる

最後に連続的な可視化の様子を示す。

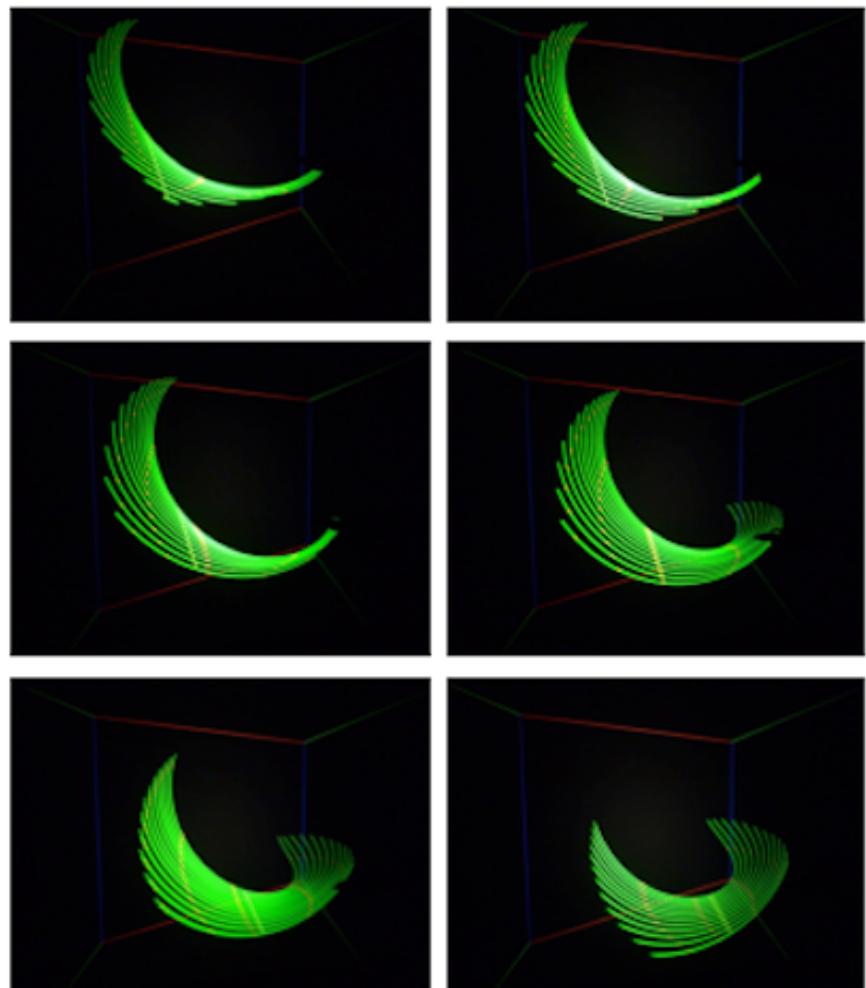


Fig. 17: A series of snapshots of the visualization applied to a geodynamo simulation. 左上の画像から順に、ワンドを前方に向けたまま弧を描くように移動させている

5　まとめ

本研究では、CAVE型VR装置を利用した流れ場の新しい3次元可視化手法を提案し、それを実現する可視化プログラムを開発した。本プログラムはCAVE型VR装置用の汎用可視化ソフトウェアVFIVEのフレームワークに組み込んだ。そして地球ダイナモシミュレーション結果についてこのプログラムを適用し、力線がリアルタイムに可視化されることを確かめた。本手法は流れ場の力線を、対話的に選択された任意の場所で、連続的かつリアルタイムに表示するものである。本手法を用いることにより、大規模な3次元ベクトル場中の、流れ場を確かめたいある部分の局所的解析が効率的に可能になると期待できる。より効率的な可視化のために、直線状の出発点以外にも、例えば平面的に出発点を与えるよう切り替えられるようにするなどの改良が考えられる。

6　謝辞

本研究にあたり、指導を賜りました指導教官陰山教授に感謝いたします。また、粘り強く相談に乗って下さいました政田助教授に感謝します。同研究室の目野氏と山浦氏には、研究に関わる多くの助言をいただきました。

参考文献

- [1] Brian Cabral, Leith C. Leedom, “Imaging Vector Fields Using Line Integral Convolution”, Proceedings of the 20th annual conference on Computer graphics and interactive techniques, pp. 263-270, (1993)
- [2] 可視化情報学入門編集委員会, “可視化情報学入門”, 東京電機大学, pp170-173, (1994)
- [3] Lloyd A. Treinish, Zaphiris D. Christidis, “Visualization Techniques for Applications of High-Resolution Numerical Weather Models”, Proceedings of the Sixteenth International Conference on Interactive Information and Processing Systems for Meteorology, pp. 66-69, (2000)
- [4] OpenGL策定委員会, 松田晃一訳, “OpenGLプログラミングガイド 原著第5版”, ピアソン・エデュケーション, (2006)

- [5] 陰山聰, 大野暢亮, “バーチャルリアリティを用いた対話的 3 次元可視化ソフトウェアの開発とその応用”, プラズマ・核融合学会誌, Vol.84, No.11, pp.834-843, (2008)
- [6] N. OHNO, A. KAGEYAMA and K. KUSANO, “Virtual reality visualization by CAVE with VFIVE and VTK”, J. Plasma Physics Vol.72, part 6, pp.1069-1072, (2006)
- [7] 大野暢亮, 陰山聰, “CAVE 用汎用可視化ソフトウェア VFIVE ”, 第 24 回 CG・可視化研究会 (CAVE 研究会), (2006)
- [8] プラントル, “流れ学 (上)”, コロナ社, pp36-38, (1972)
- [9] 小島紀男, 町田東一, “FORTRAN 基礎数値計算”, 東海大学出版会, pp209-211, (1989)
- [10] 陰山聰, “コンパスはなぜ北を指すのか?”, 科学, Vol.77, No.5, pp.532-538, (2007)