

地球シミュレータデータの可視化

陰山 聡、大野 暢亮

海洋研究開発機構

地球シミュレータセンター

1 はじめに：データの可視化とは

NKH ラジオの「気象通報」という番組を御存じだろうか？各地の風向、風力、気圧などの数字をアナウンサーが次々に読み上げていく番組である。これをただ耳で聞いているだけでは天気は分からない。しかしペンと地図を手にし、番組を聞きながら読み上げられる各地点に気圧などを書き込み、最後に同じ気圧になりそうな点を曲線（等高線）でつなげていくと、低気圧や高気圧が図—天気図—としてあらわれる。知識のある人が天気図を見れば、明日の天気がだいたいわかるらしい（私にはできないが）。

この天気図に入っている情報は、もともとラジオのアナウンサーが読み上げる数値データに全て含まれていたはずである。だが、そのような数値データの羅列をいくら注意深く聞いてもあるいはそれを数表にして紙の上に書き、それをじっと睨んでみても明日の天気を知るのは難しい。紙の上に天気図という図にしてみればじめてデータに隠されていた情報を引き出すことができたのである。このようにたくさんの数値データを何かの方法で図にする操作が可視化である。天気図はデータ可視化の一番身近な例であろう。

天気図の場合は観測されたデータであったが、同じことは計算機シミュレーションの出力データについても言える。計算機でシミュレーションを行い、最後に出てくるもの（出力）はやはり数字の羅列である。気象通報の場合には、観測データが得られる場所（アナウンサーが読み上げる地名）の数は、日本各地とその近所で、せいぜい 50 から 60 地点程度である。それにその数字といっても「風力 3」とか、「1029 ヘクトパスカル」などという一桁や、せいぜい四桁程度の整数なので、それほど大変な量の数値データではない。ところが、計算機シミュレーション、特に地球シミュレータを使って行うような大規模な計算機シミュレーションになると、データがとれる位置—つまり計算格子点—の数は、後述するように 10^9 (10 億) 点に上ることも珍しくない。また、その数字も整数ではなく、ふつうは一つの数字あたり 8 バイトの情報量で表現される実数（倍精度実数）である。（これは 10 進の小数で書くと 16 桁近くある。）このような膨大な量の数値データになると、紙とペンを使って等高線を描くことなど不可能である。そもそも現在のシミュレーションは天気図のような 2 次元的なデータを扱うことはまれであり、現実の（3 次元の）複雑な現象を 3 次元のまま計算したものが多い。今日のシミュレーション研究者は、単に膨大なサイズのデータを処理するというだけでなく、その数値データの海の中に潜む複雑な 3 次元構造を可視化を通して抽出し、理解しなければならない。

2 シミュレーション科学における可視化の重要性

計算機シミュレーションを研究手段とする科学、すなわちシミュレーション科学は、二つの技術的な柱に支えられている。一つは計算の技術すなわちコンピュータであり、もう一つは可視化に関する技術である。コンピュータの性能は現在でも指数関数的な進歩を続けているが、一方もう一つの柱である可視化技術の方はそれに十分追いついていないのが現状である。

大昔、計算機の性能がまだ貧弱であった時代、対称性を仮定することでシミュレーション研究の対象の次元を 1 次元や 2 次元の落として計算していたころは、可視化技術も比較的単純であった。紙の上やコンピュータのモニタ画面に映し出された簡単なグラフや、等高線などを描けば、計算結果を理解するのに十分であった。これは天気図という 2 次元的な可視化手法で天気が理解できるのと同じである。

その後、計算機の性能が進歩して 3 次元シミュレーションが当たり前となってきた。我々が大学院生であった 1990 年ころには、合計 $100 \times 100 \times 100$ 程度のオーダーの格子点数を使った 3 次元シミュレーションも珍しくなかった。このような規模のシミュレーションデータの可視化になると紙やモニタ画面を使った 2 次元的な可視化手法ではまにあわない。そのため新しい可視化技術がそのころ普及しはじめた。それはグラフィックワークステーションと「3 次元可視化ソフト」と呼ばれる市販の可視化専用ソフトウェアである。この 3 次元可視化ソフトを使えば、 $O(100^3)$ の格子点に分布した様々な物理量の等値面や、断面図など、さまざまな可視化手法を組み合わせた可視化プログラムを簡単に作成することができ、さらにそうしてできた可視化オブジェクト（複数の断面図や等値面など）をマウスを使って自由に回転させたり、拡大・縮小したり、あるいは照明や色づけのしかたを変えることができる。

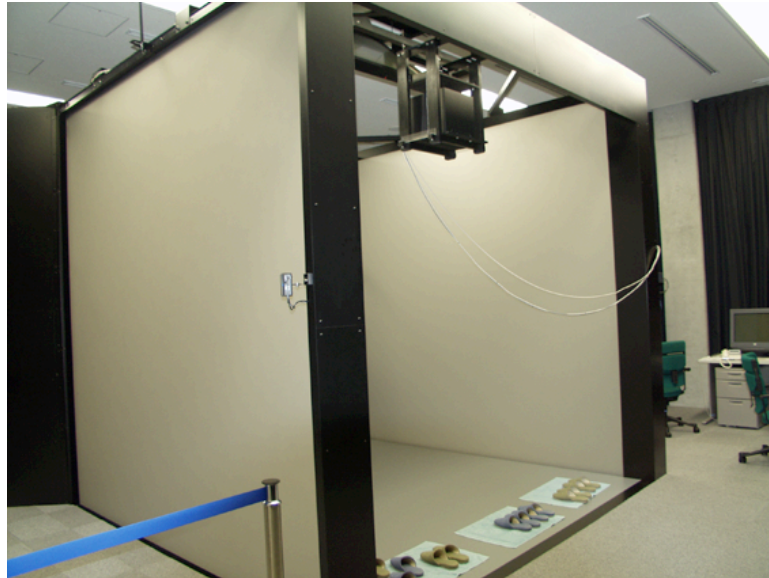


図 1: 地球シミュレータセンターに設置された CAVE 型 VR 装置 “BRAVE”。

計算機の性能がさらに向上した現在、その最先端をゆく地球シミュレータでは $O(1000^3)$ の出力データを日常的に可視化し、解析しなければならない。この規模になると、従来の可視化手法はもう限界である。確かにこの間、ワークステーションやパソコンのグラフィック性能も向上したが、いくら大量のポリゴンを高速で回転することができても、その可視化オブジェクト（たとえば等値面）そのものが本質的に複雑な幾何学的 3 次元形状をもっている場合、現象の理解にはあまり役に立たない。あきらかに可視化技術の質的な向上が必要とされている。

さらに従来の可視化手法にはもう一つ問題がある。それはベクトル場の可視化に弱いという点である。計算機シミュレーションの結果得られる出力データはスカラー場であるとは限らない。 $O(1000^3)$ の格子に分布した複数の 3 次元ベクトル場（たとえば流体の速度場、電場、磁場など）の 3 次元的な構造を直感的に把握しなければならない。ベクトル場の可視化は非常に困難だが、将来ぜひとも克服しなければならない課題である。

シミュレーション科学を支える二つの技術的な柱（計算と可視化）にアンバランスが生じている現在、これを解消するためには新しい基本技術の導入が必要である。我々はその有力候補は仮想現実（Virtual Reality = VR）技術だと考えている。現在、さまざまな方式の VR 装置が開発されているが、そのなかでも CAVE と呼ばれる方式の VR 装置が我々の目的—VR 技術を応用して 3 次元シミュレーションデータを可視化・解析するという目的—には最も適していると考えている。CAVE は 1990 年代はじめ、米国イリノイ大学シカゴ校において開発された [1]。地球シミュレータセンターでは 2003 年に CAVE 装置を導入し、これを BRAVE と名付けた [2]。

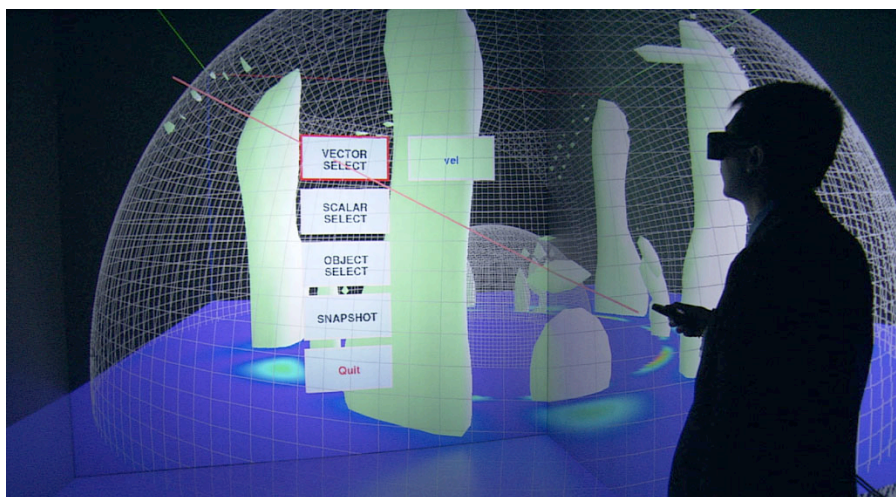


図 2: VR 可視化ソフト VFIVE。VFIVE ではこの図のように、目の前に浮かんだメニューを仮想レーザービームで選択することにより可視化手法を切り替える。

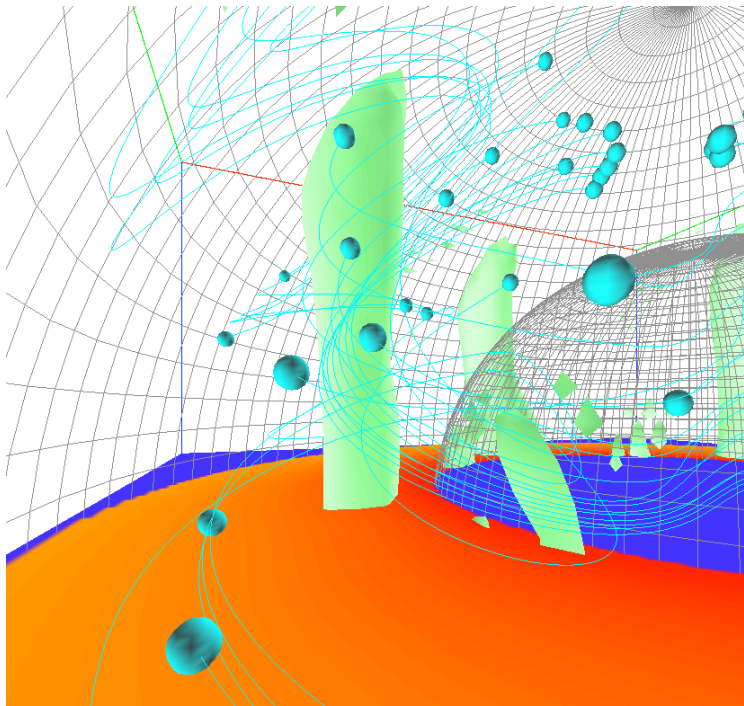


図 3: VFIVE の可視化例。手の先から次々とテスト粒子を放出し、その運動を観察することで 3 次元の流れ場構造が直感的に把握できる。

3 VR 可視化のハードウェア：BRAVE システム

BRAVE の中心部は一辺がちょうど 3m の立方体の部屋である (図 1 参照)。この部屋の 4 つの面 (3 つの壁と床) はステレオ映像が投影されるスクリーンになっており、データを解析するシミュレーション研究者 (ここではビューアと呼ぶ) は立体眼鏡¹をつけて、この部屋の中に入る。ビューアはステレオ画像に文字通り取り囲まれるので、常に広い立体視野角が保証される。(つまりきょろきょろと見回してみても常に目に立体画像が入ってくる。) これが CAVE 装置で非常に高い没入感が得られる重要な要因である。スクリーンどうしの境目でもステレオ画像が互いに滑らかにつながるように投影される。可視化物体のコントロール、つまり VR 世界とのインターフェースにはワンドと呼ばれる手持ちのコントローラ (一種の 3 次元マウス) を使う。眼鏡とワンドには磁気センサーがついており、ビューアの視点の位置・視線方向や、手の位置・向きをリアルタイムで検出している。ビューアが BRAVE 内部を自由に歩き回ったりしゃがみ込んだりしても、常にその位置から見えるべき映像がリアルタイムで 4 つのスクリーンに投影されるので、全てが自然に見える。たとえば目の前に何か仮想物体 (ボール) が浮いているとして、その向こう側を見たければ、自分の足で歩いて行ってボールの向こう側に回り込み、振り返ってみればいい。リアルタイムで視点の検出をしていることと、それに基づいて自動的に投影画像が調整されること、通常の立体映画と CAVE システムの大きな違いである。

4 VR 可視化ソフトウェア：VFIVE

我々は以前から CAVE システム用の可視化プログラムを開発してきた [3, 4]。また、我々は一般の CAVE システムを用いて OpenGL と CAVE lib を使って VR プログラムを作るための入門的なプログラミングガイド [5] も書いた²。これまでさまざまな可視化プログラムを書いてきたが、現在もっとも力を入れているのは、任意の CAVE システムに利用できることを目指した汎用のシミュレーションデータ可視化プログラム VFIVE (図 2) である [6]。現在 VFIVE の最新版は v3.8 である³。

VFIVE には等値面表示、断面図表示などの基本的な可視化手法に加えて、BRAVE (あるいは一般の CAVE 装置) の対話的な操作能力を有効に活用した独自の可視化手法を実装している。たとえば localArrows という可視化手法では、ワンドの周囲に立体的な矢印群があらわれ、各矢印の向きと長さによってその点のベクトル場が分かる。これだけであれば特に珍しい可視化手法ではないが、localArrows では、ビューアがワンドを動かすと、矢印群がその手の動きについて動き、このとき各点でのベクトル場の向きと強さをリアルタイムで補間しながら

¹液晶を使ったシャッターつきの眼鏡。右と左に交互に高速にシャッターがおりる。右目用の画像がスクリーンに投影される瞬間、それに同期して左目にはシャッターがおりるので、常に右目は右目用の画像を見る、左目は左目用の画像を見ることで画像が立体視を実現する。

²残部あり。希望者は kage@jamstec.go.jp までどうぞ。

³最終版は v.5 と決めている。

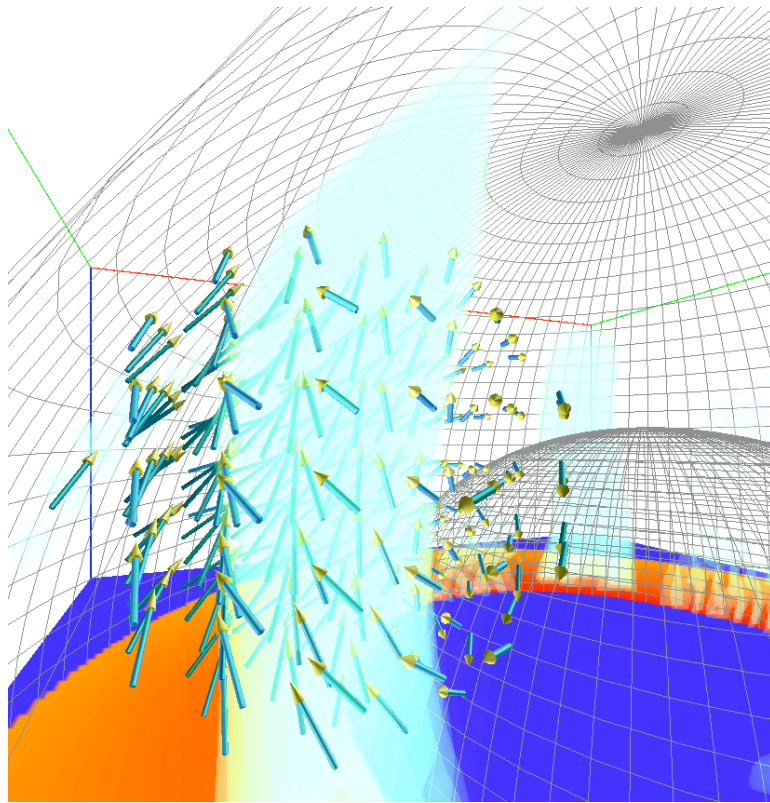


図 4: VFIVE の ver. 3.8 に組み込まれたボリュームレンダリングの表示例。3次元テクスチャマッピングを利用することにより、CAVE 内でのリアルタイムボリュームレンダリングを実現した。

表示するので、注目する場所の近くで、「手を上下左右に動かしつつ矢印群の向きと長さのダイナミックな変化を観察する」ことによってそこでのベクトル場の構造を直感的に把握することが可能となる。これは複雑なベクトル場の解析には非常に有効な可視化手法である。

また VFIVE には力線追跡やテスト粒子の放出機能もある。これは市販の可視化ソフトにもよくあるベクトル場のごく標準的な可視化手法であるが、力線やテスト粒子の出発点をうまく指定するのに苦労した経験を持っている人も多いに違いない。我々の VFIVE ではそれが直感的に指定できる。ワンドを持った手を好きな位置まで伸ばし、そこでワンドのボタンを押すと手の先からテスト粒子が次々と放出される。そして、そのテスト粒子が流れ場に乗って飛んでいる運動の様子を立体的に、好きな角度から観察することができる (図 3)。ある粒子が少し流れていった先で突然向きを変えたりするなど、なにか興味深い運動を示したら、その場所まで歩いて行って手を伸ばしそこでまた集中的にボタンをなんども押すことで新たな粒子群を放出させてその運動を観察すればいい。このような可視化手法は CAVE 装置でなければできない種類の、つまり VR の特徴をうまく生かした可視化手法であると言えると思う。

雲のようにぼんやりとした物理量や、一つのせまい領域に極値が極端に集中した物理量の分布を可視化する場合、等値面表示は適切な可視化手法ではない。このような場合、ボリュームレンダリング [7] と呼ばれる可視化手法が有効である。ボリュームレンダリングとは、与えられたスカラー場を値に応じた半透明な色の場に置き換え、設定した視点 (一点) と、投影面 (2次元面) 上の各画素とを結ぶ多数の直線 (光線) 上でその色を積分する手法である。この定義から分かるとおり、ボリュームレンダリングは 1 枚の画像を生成するだけでも膨大な計算が伴う。

CAVE ではビューアの視点の移動に反応してリアルタイムで画像を更新する必要がある。つまり、(i) 右目用と左目用の二つの画像を; (ii) 4つの投影面 (3つの壁スクリーンと床スクリーン) それぞれに対し; (iii) 毎秒最低数枚、できれば 20 枚以上; のボリュームレンダリング画像を生成できなければ CAVE では使えないということになる。この意味でボリュームレンダリングは「重すぎて」CAVE で実現することは難しい。しかし最近のグラフィックワークステーションの一部にハードウェアとして実装されている 3次元テクスチャマッピングと呼ばれる機能をうまく活用すれば、ボリュームレンダリングを非常に高速に計算することができる。我々はこの高速ボリュームレンダリングを BRAVE で実現し、最近 VFIVE に組み込んだ (図 4)。VR システムでの Volume Rendering (Virtual Reality Volume Rendering, VRVR) は VR 可視化において現在、活発に研究されているテーマである [8]

最近 VFIVE に加えたもう一つの大きな改良は、可視化ツール VTK⁴ (Visualization Tool Kit)[9] の組み込みである。VTK はオープンソースの可視化ツール群で、その注意深い設計と多彩な可視化機能の充実ぶりは群を抜

⁴<http://public.kitware.com/VTK/>

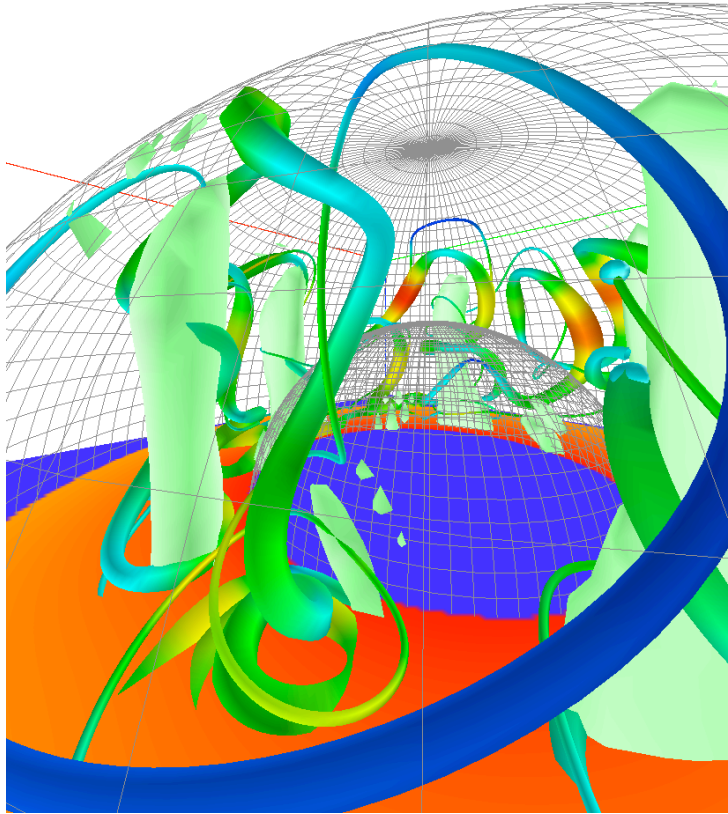


図 5: VTK の Stream tube 機能を組み込んだ VFIVE v3.8 の可視化例。tube の直径と色はその点での速度ベクトルの強さを示している。

いている。VTK は主に PC を使った手軽な、汎用の可視化ツールとしての利用が想定されているようであるが、それぞれの可視化機能は独立したモジュール構造になっており、一方、VFIVE の可視化機能も同様にモジュール構造としてデザインされているため、VTK の可視化機能を VFIVE に取り入れることは自然であり、難しいことではない。図 5 と 図 6 に VFIVE v3.8 に組み込んだ VTK の可視化機能二つの例を示した。

5 最後に

計算機の演算性能がいくら高くなっても計算結果を効率的に可視化し解析できなければその価値は半減してしまう。近い将来、夢のような超々高速シミュレータが我々の手に入ったと想像しよう。そして現在では夢のような超々大規模シミュレーションをたった一晩で計算できてしまうでしょう。だが、もしもその計算結果を解析するのに 1 ヶ月かかるとしたら、それは実に馬鹿げたことではないだろうか？シミュレーション研究の発展のためには我々は可視化技術の革新に真剣にとりくむ必要がある。本稿ではその有力候補として VR 技術の利用について述べた。

我々が必要としている可視化技術は、可視化の効率を上げる技術である。その意味で上に述べた“可視化技術の革新”という表現は少々大げさ過ぎるかもしれない。「CAVE のような VR 装置を使わなければ絶対に不可能である」というような可視化の例は数は少なく、あるとしても例外的であろう。重要なのは「VR 技術を使えば可視化を格段に効率化することができる」ということである。VR 以前の可視化技術（PC モニタと市販の 3 次元可視化ソフトの組み合わせ）を使っても、上に書いたような超々大規模シミュレーションのデータを可視化し、解析することは不可能ではないであろう。あるいはもっと古い可視化手法—単純な断面図やあるいは 1 次元のグラフプロット—であっても、無数の断面を試し、それを頭の中で 3 次元的に再構成すれば得られたシミュレーション結果を理解することは時間さえかえれば不可能ではないであろう。問題はそれがあまりにも効率が悪いという点である。先端的なシミュレーション研究者がいま痛切に必要とするものは、大規模なデータを効率的に可視化し表現する「道具」である。そのような道具はシミュレーション研究者が自分で作るか、あるいはシミュレーションデータの可視化手法・表現手法を専門とする研究者が、シミュレーション研究者と密接に連絡をとりながら開発していく必要がある。

地球シミュレータセンターの高度計算表現法研究グループでは、このような認識のもと、本稿で述べた VR 可視化ソフトの開発以外にも、大規模なシミュレーションデータを一晩で動画化するツールの開発や、多数の分散ファイルを異なった解像度で統合的に可視化処理するツールの開発、あるいは流れ場を可視化するための新しい

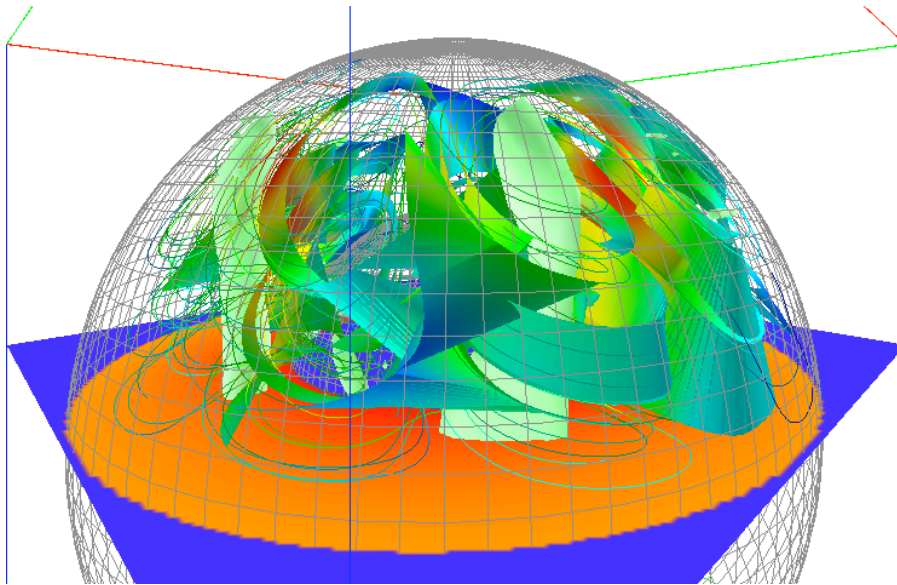


図 6: VFIVE v3.8 に組み込んだ VTK の可視化機能のもう一つの例 (Stream surfaces)。

可視化アルゴリズムの開発などに精力的に取り組んでいる。

参考文献

- [1] Cruz-Neira, C., Sandin, D. J., and DeFanti, T.A., Surrounded-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE, *ACM SIGGRAPH 93*, 135–142, 1993
- [2] 荒木文明, “仮想地球” の可視化とその表現, *情報処理*, **45**, No. 2, 22–26, 2004
- [3] A. Kageyama, T. Hayashi, R. Horiuchi, K. Watanabe, and T. Sato, Data Visualization by a Virtual Reality System, Proceedings of 16th International Conference on the Numerical Simulation of Plasmas, pp.138-142, Santa Barbara, CA, USA, 10-12 February 1998.
- [4] Kageyama, A., Tamura, Y., and Sato, T., Scientific Visualization in Physics Research by CompleXcope CAVE System, *Trans. Virtual Reality Soc. Japan*, **4**, 717–722, 1999.
- [5] 陰山 聡, 佐藤 哲也 VR システム CompleXcope プログラミングガイド, Research Report NIFS-MEMO No.28 (1998) pp.1-115.
- [6] Kageyama, A., Tamura, Y., and Sato, T., Visualization of Vector Field by Virtual Reality, *Progress Theor. Phys. Suppl.*, **138**, 665–673, 2000.
- [7] Drebin, R. A., Carpenter, L., and Hanrahan, P., Volume Rendering, *Computer Graphics*, **22**, 65–74, 1988.
- [8] Suzuki, Y., Sai, K., Ohno, N., and Koyamada, K., 球面サンプリング版ボリュームレンダラーの研究・開発, 第 32 回可視化情報シンポジウム講演集, Vol.24 Suppl., No. 1, pp. 443-446, 2004
- [9] Schroeder, W., Martin, K., and Lorensen, B., *The Visualization Toolkit*, Third Edition, Kitware Inc., New York, 2002.