

数値風洞におけるレーザーライトシート法の実現[†]

上 沢 和 真*・小山田 耕 二**・土 井 章 男**

ABSTRACT In a wind tunnel, a laser light sheet method is often used to identify vortices and visualize them in a flow field. To make it easy to compare an experimental result with a numerical result, it is crucial that we should develop a visualization technique that is related to the laser light sheet method. In this paper, we propose two techniques, LIC based and physically based, which visualize the flow field near the vortex. We clarify that the former technique is superior to the latter in the viewpoint of computational cost by using several flow fields.

1. はじめに

渦は、流体運動の最も基本的な運動形態のひとつであり、流れの力学機構や制御の考察には欠かせない概念である。実際、カルマン渦、ハリケーン、航空機の翼端渦、壁乱流の流れ方向渦等々、渦構造が流体力学的に重要な働きをしている現象は数多い。この渦構造を解析しそのダイナミクスを理解するためには、流れの中に現われる渦を個別に同定し、可視化できることが望ましい^⑧。

複雑な流れを解明するためにしばしば用いられる風洞では、任意の断面での流れを可視化するため、レーザーライトシート法が用いられる。レーザーライトシート法では、模型の風上で煙や微粉末などを風洞内に発生させ、レーザーを平面状に照射する。名古屋大学航空学科低速風洞では、水の微粒子をトレーサとして渦付近だけに散布し、渦軸に垂直なレーザーライトシートを用いて、デルタ翼面上の前縁剥離渦を可視化している（図1参照）^①。

風洞実験との比較を容易にするためには、数値風洞における可視化手法としてレーザーライトシート法に対応したものが有効である。数値風洞では、3次元格

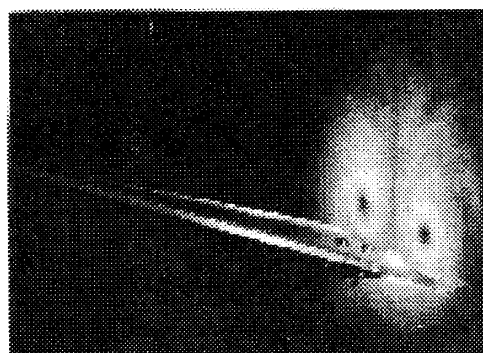


図1 レーザーライトシート法画像例

子点で速度データが出力されている。レーザーライトシート法を模倣するには、渦軸に垂直な断面上で速度データをマッピングし、2次元平面上で定義された速度場（2次元速度場）を生成する。平面全体にわたって2次元速度場を可視化する手法は、商用可視化ツールに多く見られる^⑥。しかし、レーザーライトシート法では、渦から離れていくとトレーサーが次第に消滅していくので、断面全体を可視化対象とするのではなく、渦付近だけを局所的に可視化する手法の開発が重要となる。我々は、この可視化方法を開発するために2つのアプローチを採用した。

ひとつは、2次元速度場の可視化手法として提案されているLIC法^⑤をベースとするアプローチである。LIC法は、ホワイトノイズ画像を入力とし、ベクタ場から計算される局所流線に沿った畳み込み積分計算を行い、ベクタの方向に対してコヒーレンスのある画像を生成する。LIC法は速度場の全体的把握に有用であるが、渦付近だけに表示領域を限定した手法は報

Development of Lasor Light Seats fof Virtual Wind Tunnel.
By Kazuma Kamisawa (Computer and Information Science, Iwate University), Koji Koyamada and Akio Doi (Faculty of Software and Information Science, Iwate Prefectural University).

*岩手大学工学研究科

**岩手県立大学ソフトウェア情報学部

†2000年3月2日受付 2000年6月19日再受付

①画像提供：MATSUNO Takashi, Fluid Dynamics Lab, Aerospace Eng, Nagoya Univ

告されていない。

もうひとつは、渦付近に仮想トレーサーを配置し、2次元速度場に添って広がるトレーサーの濃度分布を物理シミュレーション（移流方程式）を用いて計算することである。各画素で計算された濃度値は、画像テクスチャにおける輝度値に対応させる。移流計算を行うために *Patankar* の提案する有限体積法¹⁾を利用する。ここで2次元速度場は最初から与えられているものと仮定することができ、速度場の計算は必要ではない。

本論文第2章では、3次元格子点で定義された速度データから、渦中心を通過し渦軸に垂直である平面で速度場を切り出す手法について述べる。第3章では、渦付近だけに表示領域を限定した LIC 手法について、そして第4章では、移流方程式を用いたトレーサーの濃度分布の計算手法について説明する。第5章では、これら2つの手法を3次元空間で定義された速度データに適用し、計算時間、画質の観点で評価を行う。また、WWW ベース可視化システムへの本手法の実装についても述べる。

2. 速度テクスチャの生成

本論文で提案する可視化手法では、渦中心を通過し、渦軸に垂直なスライス平面上に速度ボリュームデータをマッピングして、速度テクスチャを生成する（テクスチャ付き4角形の作成法については3章以下で述べる）。そのためにボリュームデータからスライス平面を3角形メッシュとして表現し、その3角形メッシュをスキャンコンバージョン⁹⁾することにより画素点における速度データを計算する。ボリュームデータからスライス平面を3角形メッシュとして表現するには、スライス面と格子稜線との交点を計算し、交点を頂点とする3角形メッシュを作成する。頂点では格子点で定義されたベクタデータを用いて補間計算を行う。スライス面上で定義される3角形の頂点は、スライス面と格子の稜線との交点である。その数は、格子の数の $2/3$ 乗に比例し、数十万の格子からだと数万個程度となる。我々はインターネットを通じて後に述べる画像テクスチャを転送しようとしているので3角形の数が数万というのは問題である。この問題の解決策については⁹⁾で述べられている手法、すなわち3角形メッシュとして表現されたスライス面形状を1個のテクスチャ付き4角形として表現する手法を採用している。

3角形メッシュをスキャンコンバージョンするに

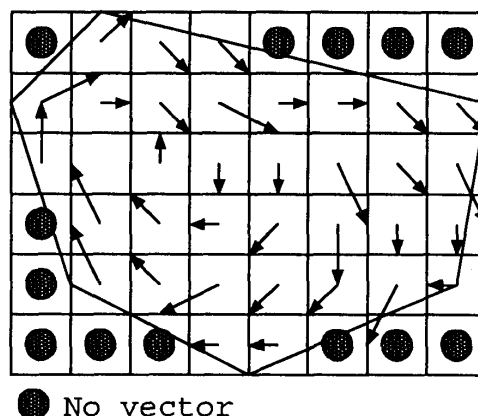


図2 速度テクスチャ

は、メッシュの座標点に対しスライス面の法線、すなわち渦軸と平行な視線を仮定して視点座標系への変換を行う。そして、引き続きスクリーン座標系への変換が行われる。この変換は、速度テクスチャ生成のために必要となるものであり実際の視点を考慮する座標変換は、VRML ブラウザ内部で行われる。各画素点における速度データ値は、三角形の頂点で定義された速度データを用いた補間計算により決定される。各データ値は、あたかもカラーデータであるかのようにフレームバッファに書き込むことも可能である。この場合各画素で表現できるデータの大きさの上限を調べ、その結果に基づき量子化を行う必要がある。スキャンコンバージョンを行った後、スライス面を表現する4角形を定義するためにスクリーン上でスキャンコンバージョンの対象となった画素領域を囲む最小面積をもつ長方形（画像テクスチャ）を計算する（図2参照）。

3. 移流方程式を使用した手法

スキャンコンバージョンにより各画素の中心点で速度ベクタが定義された速度テクスチャを生成することができた（図2参照）。但し、透過属性が与えられているところでは、速度ベクタはゼロとなっている。本章では、移流方程式を使って、この速度テクスチャから画像テクスチャを生成する方法について述べる。本論文では、テクスチャで定義された画素を計算用格子として使用する。

3.1 移流計算の基本スキーム

速度テクスチャは、2次元格子で定義された速度データである。2次元格子上における渦中心の近傍にトレーサの供給源を配置し、そこから発生するトレーサが速度場に仕掛けてどのように分布していくのかを

計算する。トレーサの質量を無視するので、トレーサの分布はベクタ場を表現すると考えることができる。トレーサの濃度分布をピクセル値とするような画像テクスチャを作成すれば直観的な可視化画像を生成することができる。

ここでは、一定時間後のトレーサの濃度分布を計算することを考える。ここで述べる計算手法は、文献1)を参考にした。2次元ベクタ場を (u, v) そしてトレーサの分布を各格子での濃度値 ϕ で表現すると、 ϕ は、次に示す移流方程式の解となる。

$$\frac{\partial \phi}{\partial t} + \frac{\partial (u\phi)}{\partial x} + \frac{\partial (v\phi)}{\partial y} = S \quad (1)$$

ここで、 S はトレーサの生成速度を表す。質量保存の法則は、次のように記述できる。

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (2)$$

式(1)、式(2)は、各計算格子毎で成立するので、それらを格子の各表面（格子の表面を以降界面と呼ぶ。）に対して積分すれば、

$$a_P \phi_P = a_E \phi_E + a_W \phi_W + a_N \phi_N + a_S \phi_S + b \quad (3)$$

となる。各係数 a_P , a_E , a_W , a_N , a_S , b は、格子における速度データから算出される。詳細は、参考文献1)を参照してほしい。式(3)は各格子で成立し、したがって、格子の数だけの連立方程式ができる。これらの連立方程式を解けば、格子毎にトレーサ濃度値が定義される。これらのトレーサ濃度は、物理法則を満足するので、本手法は物理的に直観的な画像テクスチャを生成することになる。

3.2 偽拡散の低減

式(3)は、隣接する4つの格子からの寄与だけを考慮していたので格子線と平行しないベクタ場では偽拡散が発生する¹⁾。偽拡散とは、方程式上で拡散効果を仮定していないにもかかわらず、流れの方向が格子の界面と平行していない場合、あたかも拡散効果を考慮したような計算結果が得られる現象のことをいう。Becker²⁾は、流れの流入口に座標そのものの移流計算を行い、その座標上で畳み込み積分を行い、流れ場を可視化している。この場合、偽拡散を減少させるためにもとの格子をさらに16分割して、上流法をベースとした有限要素法を適用した。流入された座標がすべての空間座標をうめ尽くすまで計算を行わなければならない。膨大な計算量が必要となることが問題となる。

われわれは、偽拡散を解決するために図3に示すように格子の各頂点を含む格子線をもとの格子から45°回転させた格子を考え、ここからの流量も考慮する。

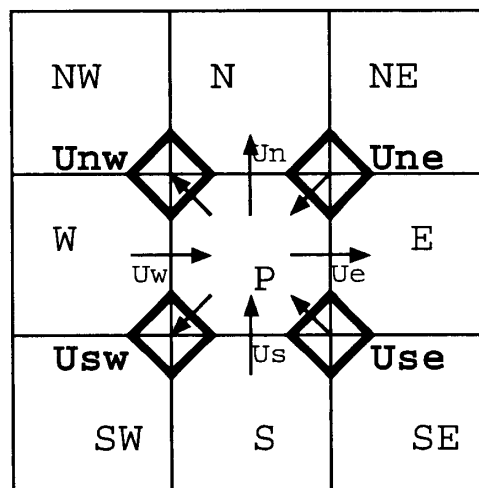


図3 新しく配置する格子

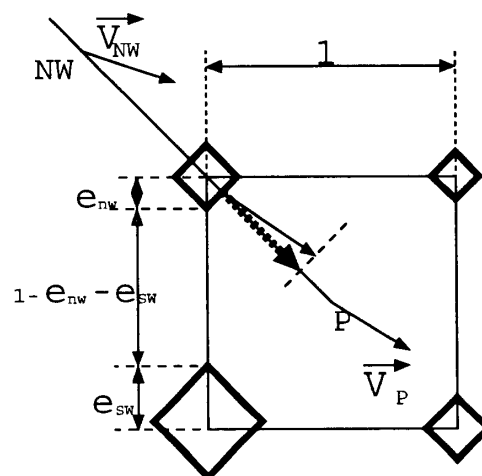


図4 界面における補間

この流量は、2つの格子で定義されたベクタデータから計算する。例えば新しく導入した格子（以降45度格子と呼ぶ）の界面に垂直方向の速度 u_{nw} は、斜め方向に位置する格子と着目する格子の中心における速度データを使って補間計算する（図4を参照）。前者を $\bar{V}_{NW} = (u_{NW}, v_{NW})$ 、後者を $\bar{V}_P = (u_P, v_P)$ とすると、

$$u_{nw} = \frac{(\bar{l}_{nw}, (1 + \epsilon_{nw}) \times \bar{V}_P + (1 - \epsilon_{nw}) \times \bar{V}_{NW})}{|(1 + \epsilon_{nw}) \times \bar{V}_P + (1 - \epsilon_{nw}) \times \bar{V}_{NW}|} \quad (4)$$

となる。ここで、 $(\bar{A}, \bar{B})$ は、ベクタ A とベクタ B の内積を表し、また、 $\bar{l}_{nw}$ は、NWの逆方向の単位ベクタ $(1/\sqrt{2}, -1/\sqrt{2})$ である。他の3方向についても同様である。ここで、 ϵ_{nw} は、45度格子の大きさを表現するパラメータであり、その値は格子の稜線の長

さに対する相対値を意味する。45度格子同士が重なり会わないように ε の有効な値の範囲は、0から0.5とし、その値は、格子へ方向ベクタ（この場合NW）からのベクタデータのずれ角 θ を使って以下のように計算する。

$$\varepsilon_{nw} = 0.5(2 \cos^2 \theta - 1)^N \left(\theta \leq \frac{\pi}{4} \right) \quad (5)$$

$$= 0 \left(\theta > \frac{\pi}{4} \right) \quad (6)$$

ここで N は、ずれ角に関する指向性を表現する非負の整数であり、値が大きくなれば鋭い指向性を持つことになる。本論文で、方向NWに対する指向性とは、流れの方向がNW方向に近いほどより多くの流量を考慮するという意味で使用する。

式(1)、式(2)をを格子の全界面に対して積分し、格子中心で定義された濃度値 ϕ_P についてまとめると

$$a_P \phi_P = a_E \phi_E + a_W \phi_W + a_N \phi_N + a_S \phi_S + a_{NW} \phi_{NW}$$

$$+ a_{SE} \phi_{SE} + a_{NE} \phi_{NE} + a_{SW} \phi_{SW} + b$$

$$a_E = A(-u_e, 0)(1 - \varepsilon_{ne} - \varepsilon_{se})$$

$$a_W = A(u_w, 0)(1 - \varepsilon_{nw} - \varepsilon_{sw})$$

$$a_N = A(-u_n, 0)(1 - \varepsilon_{nw} - \varepsilon_{ne})$$

$$a_S = A(u_s, 0)(1 - \varepsilon_{sw} - \varepsilon_{se})$$

$$a_{NW} = A(-u_{nw}, 0)(\sqrt{2} \varepsilon_{nw})$$

$$a_{SE} = A(u_{se}, 0)(\sqrt{2} \varepsilon_{se})$$

$$a_{NE} = A(-u_{ne}, 0)(\sqrt{2} \varepsilon_{ne})$$

$$a_{SW} = A(u_{sw}, 0)(\sqrt{2} \varepsilon_{sw})$$

$$a_P = a_E + a_W + a_N + a_S + a_{NW} + a_{SE} + a_{NE} + a_{SW} + \frac{1}{\delta t}$$

$$b = \frac{-\phi_P^0}{\delta t}$$

となる。ここで、大文字添字は格子の中心を、そして小文字は格子の界面を表現する。また、格子は、画像における1画素を表すから x 方向、 y 方向の格子幅は1と設定した。

文字自身の意味は、隣接関係を示し、表1に示す通りである。したがって、 u_e は、東側界面で定義され

表1 添字の意味

character	direction
E,e	+x (1,0)
W,w	-x (-1,0)
N,n	+y (0,1)
S,s	-y (0,-1)

た、界面に垂直な速度成分を表す。(図3参照) ϕ_F は、東側に隣接する格子の格子中心におけるトレーサ濃度値を示す。上添字0は、1タイムステップ前のトレーサ濃度値を表現する。関数 $A(p, q)$ は、 p, q のうちで小さくない方の値を返す関数である。この関数は、風上スキームを表現するために用いられている。風上スキームは、ある格子の界面における濃度値を、風上側の格子の濃度値に等しいと仮定する計算手法で、数値流体力学でしばしば用いられる。したがって、 $u_e > 0$ の場合、 ϕ_e は ϕ_P と等しく、そして $u_e < 0$ の場合、 ϕ_e は ϕ_F と等しくなる。

4. 領域限定型 LIC を使用した手法

一般的にLIC法はすべての画素について畳み込み計算を行う。我々は、LICによる画像計算を渦中心付近に限定したいと考えているので、各画素に対して、渦中心からの近さを定義する必要がある。特異点理論によると渦中心には、1:吹き出し、2:吸い込み、3:周回の3種類がある。これらは、速度勾配テンソルの特性方程式の解において虚数成分を調べることでより判別することができる²⁾。以下の説明では、1:吹き出しタイプの渦中心を想定する。

レーザーライト法に対応した可視化法を開発したいと考えているので、トレーサの生存時間（識別不能になる迄の時間）内にトレーサがどこまで広がっているかを定量化するための尺度を定義する必要がある。我々は、渦中心付近を出発した重さゼロの仮想粒子が各画素に到達するためにどれだけの時間を必要とするかという尺度（以下経過時間と呼ぶ）が妥当であると考え

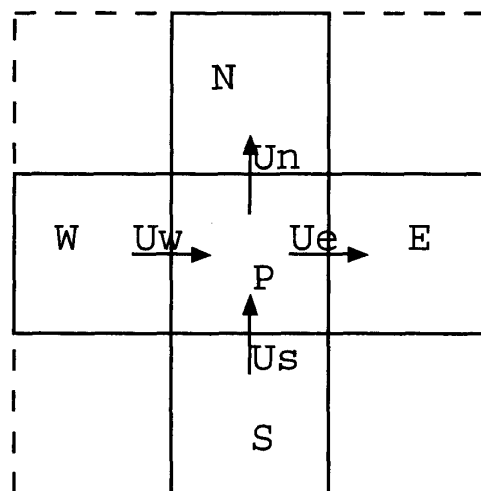


図5 基本スキーム用格子

た。各画素ごとに経過時間を計算するために、その画素点から与えられた速度場を逆方向に積分していき、画素領域境界に到達したら負の値を、そして渦中心に到達したら経過時間をその画素に対して記録する。全画素でこの記録を終えると経過時間テクスチャが完成する。

領域限定型 LIC 計算では、LIC の計算に先立ってこの経過時間テクスチャに記録されている経過時間を参照し、その値が正である時、トレーサの生存時間とその大きさを比較する。経過時間が生存時間よりも小さいとき LIC 計算を行い、そうでない場合は、計算を行わない。LIC の計算を行う場合、経過時間から決定されるトレーサ密度の減衰率を計算結果に乘じる。

経過時間テクスチャの作成には、画素の数に比例する計算時間が必要である。大部分の画素において、経過時間テクスチャに記録されている経過時間がトレーサの生存時間を上回るまたは、負の値が記入されていることになるため、これらの画素は LIC 計算の対象とならないことが予想される。このため、我々は、渦中心近傍の画素から渦中心から離れていく方向に LIC 計算を行いながら、同時に経過時間テクスチャを作成するというアプローチをとる。このアプローチを実現するアルゴリズムを以下に示す。

1. 渦中心を包含する一辺の長さが 1 である正方形の頂点に位置する 4 つの画素から LIC 計算を行う。ここで、連続する画素間の距離を 1 とする。
2. LIC 計算を行う場合、流線に沿って画素値をサンプリングすることになるが、その際、サンプリングの対象となった画素において経過時間を画素に記録する。流線は、一般的に画素の上を通過することはないので、2 画素から構成される稜線を横切るときに、その両画素に経過時間を記入する。
3. 正方形の一辺の長さを上下左右に 1 ずつ増加させる。
4. 正方形の稜線上において、ある決められた方向に（例えば時計周りに）画素を選択する。全ての画素が選択されたら、ステップ 3 に戻る。
5. 選択された画素（出発画素）において記入されている経過時間（初期経過時間）を参照する。
6. 正方形の稜線上の全ての画素において、初期経過時間がトレーサの生存時間を上回るかまたは負の値が記入されている場合、計算を終了す

る。

7. その経過時間がトレーサの生存時間より小さい場合に限り、LIC 計算を行う。LIC 計算において、サンプリングの対象となった画素において、出発画素からの経過時間に初期経過時間を加えたものを記入する。
8. ステップ 4 に戻る

以上のようにして渦近傍の画素だけを LIC 計算の対象とするような領域限定型 LIC 法を実現することができる。

5. 適用と評価

本手法は、ボリュームデータを VRML に変換する計算サーバ（可視化サーバ）に機能の一部（渦の可視化機能）として実装された（図 6）。可視化サーバは、インターネット上に設置され、ユーザの選択したボリュームデータを VRML ファイルへ変換し、ユーザへ提供する機能を持つ。

可視化サーバを用いて、移流方程式を使う手法（3 章）と LIC を用いる方法（4 章）をボリュームデータから VRML ファイルへ変換に要する計算時間と、作成された VRML ファイルとレーザーライトシートとを画質の面から比較した。

使用したデータは、 $16 \times 16 \times 16$ の 3 次元空間上で定義された同一の速度場データを用いた。速度場データの総ポリゴン数は 8000 個で対象となるのは 1616 個である。移流方程式を用いた手法（3 章）を適用した場合（図 7）、計算時間は 203.88(s) である。4 章の領域限定型 LIC を用いた手法の場合（図 8）は計算時間は 1.72(s) である。また、領域を限定しない場合は（図 9）、計算時間は 1.72(s) である。計算時間から検証すると、領域限定型 LIC を用いた手法の計算時間が

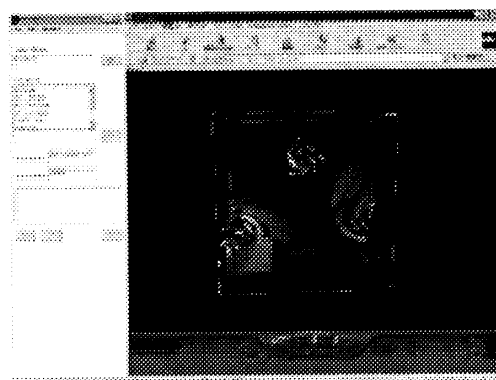


図 6 可視化結果

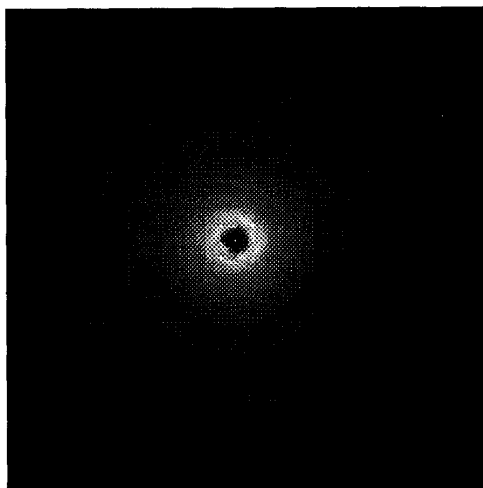


図7 移流方程式を用いた手法

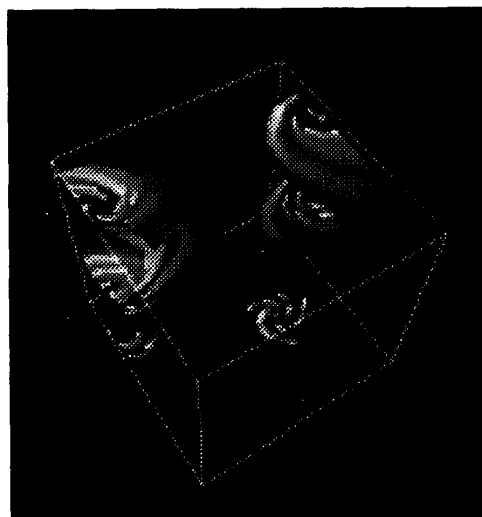


図10 複数スライス面での移流方程式を用いた手法

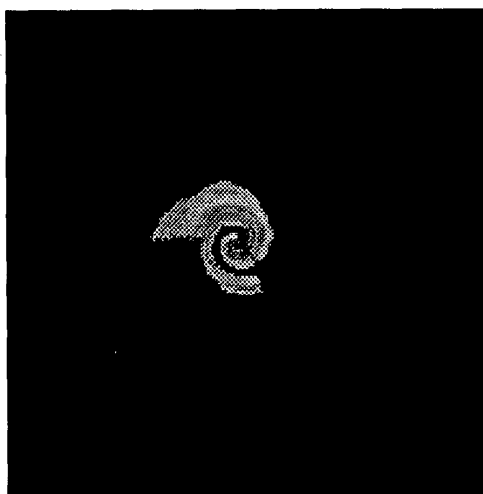


図8 領域限定型 LIC を用いた手法

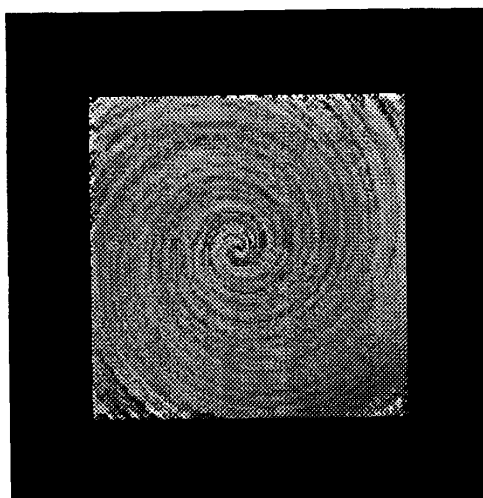


図9 領域を限定しない LIC を用いた手法

短いのがわかる。また、複数のスライス面を処理する場合には（図10）、計算時間の差が顕著である。しかし、画質的な面から考えると、レーザーライトシートに近い画質なのは移流方程式を用いた手法である。移流方程式を用いた手法は、LIC を用いた場合に比べて、向きも強さも分かりやすい。

6. 結 論

本論文では、渦の可視化についてレーザーライトシート法を模倣する手法を検討した。渦付近の速度分布を表現するために、渦中心を通過し、渦軸に垂直な平面状に速度テクスチャを定義し、2通りの方法で画像化した。得られた結果は、インターネット経由でユーザへ転送することが可能であり、また、WWWブラウザ上で簡単に3次元空間を作り出せるVRML形式とした（図6）。ひとつは、渦付近に仮想とレーサを配置し、速度場に沿って広がるとレーサの濃度分布を物理シミュレーション（移流方程式）で計算する手法であり、各画素で計算された濃度値を画像テクスチャにおける輝度値とする手法である。この手法については画質の点で優れているが計算時間が長いという問題がある。ふたつめは、LIC法により渦近傍のみを画像化すること（領域限定型LIC）である。この手法は、流線が渦の付近に戻ってくるような場合には画素の時間の再設定は行わないため、正確なLIC計算が行われない可能性があるが（図11）、移流方程式を解く手法に比べて計算速度が短い点で勝っている。将来的には2つの手法をまとめて領域限定型LICを用い

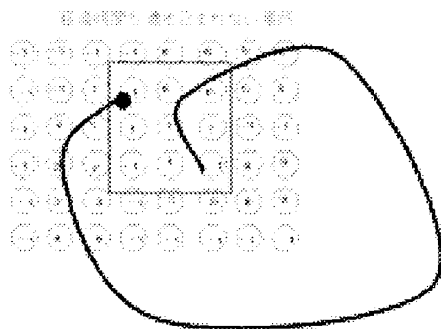


図11 書き込まれない場所

て計算領域を限定し、移流方程式を用いた可視化処理を行うことが考えられる。

参 考 文 献

- 1) スハス, V. パタンカー著 (水谷他共訳): コンピュータによる熱移動と流れの数値解析, 森北出版 (1985)

- 2) Helman J and Hesselink L: Visualizing Vector Field Topology in Fluid Flows IEEE Computer Graphics and Applications, 11-3, 36/46.
- 3) Becker J and Rumpf M: Visualization of time-dependent velocity fields by Texture Transport, Proceedings Eurographics Workshop on Scientific Visualization '98 (1998)
- 4) Koyamada K and Itoh T: Seed Specification for Displaying a streamline in an Irregular, Engineering with Computer, 14, 73/80 (1998)
- 5) Leeuw W D and Liere R: Comparing LIC and Spot Noise, Proceedings of IEEE Visualization '98 (1998)
- 6) 株式会社ケイジーテ ィ ー, AVS, <http://www.kgt.co.jp/avs/index.html>
- 7) 白山, 桑原: 仮想粒子密度とボリュームレンダリングによる流れの可視化, ながれ, 7-3 (1998), http://www.nagare.or.jp/mm/98/sirayama/index_ja.htm
- 8) 三浦, 木田: 一様等方乱流における低圧力旋回渦の同定と可視化, ながれ, 7-3 (1998), http://www.nagare.or.jp/mm/98/miura/index_ja.htm
- 9) 小山田: VRMLを用いたCFD結果の可視化技術, 機械学会論文誌