

大規模計算とボリューム・ビジュアライゼーション

白山 晋 ((株)ソフテック)

Large Scale Computing and Volume Visualization

Susumu SHIRAYAMA, SofTek Systems Inc.,
BTS-3F, 1-22-7 Sangenjaya, Setagaya-ku, Tokyo 154

This paper describes several issues about a scientific visualization system utilized in analyzing a flow field obtained by a large scale computing. Firstly, a characteristic of data by the large scale computing is considered. Secondly, we focus on the efficiency of the volume-visualization techniques, and the accuracy of a proposed method is discussed. As a preliminary case, some model fields are visualized by using a Z-slice algorithm on a personal computer system.

Key Words : Volume Visualization, Large Scale Computing

1. はじめに

CFDにおける可視化の研究は可視化技術の開発よりも応用面を中心として発展してきた。このため、実験における可視化とは異なり、基礎的な技術に関する研究は少ない。これは、次の二つの理由による。一つは、先行していたコンピュータ・グラフィックスの研究の中で開発された技術が、そのまま使えたことである。見方を変えれば、CFD固有の手法というものがある。(但し、仮想粒子の追跡法はCFD独自の可視化法として位置づけられている[1,2]。) もう一つは、ソフトウェア、ハードウェアを含めた可視化システムの理想像が、初期の段階から明示されてきたためである。このため、個々の手法よりも、統合システムの構築が重要であるとされた。更に言えば、見える流れ(出力された図)に対する吟味よりもシステム構築者のプログラミング技術が注目された時期が続いた。最近では、可視化のためのプログラミングツール(例えばAVSなど)によって見えない流れを見えるようにする労力は緩和されている。このため、可視化から解析への移行が促進されるものと思われる。しかしながら、現状では解析から可視化システムへのフィードバックが第三者の手に委ねられていることも事実として受けとめなければならない。

流れの研究の中での可視化の重要性について改めて述べる必要はないと思うが、大部分の研究者の使用している可視化システムが、そのような状況の中で成熟してきたことを思い起こす必要がある。というのは、可視化システムが次のような問題を抱えているからである。そして、それらの問題点について言及することが、本稿の目的である。

- (a) 可視化の精度
- (b) 計算環境に合った可視化手法の構築
- (c) 新しい方法の導入

(a)については、可視化手法自身の精度と計算との適合性を問題とする。後者に関して、著者は「局所補間という観点から、格子生成、計算、可視化が統合される」という予想をたて、一つの解決策を提案している[3,4]。本稿では、ボリューム・ビジュアライゼーションと局所補間ということを考えてみる。使用している可視化システムの中味を知らないという流れの研究者の注意を促したい。

(b)多くの手法が、開発時でもっとも性能のよいハードウェアを想定して修正されてはいるが、計算に適した機種と可視化に適したものは一致しない。まして、可視化手法は、流れの研究者が作ったものではないために、計算環境を考えたときに新しい可視化システムが必要となる。大規模計算を例にしてこの問題を考えてみたい。

(c)CFD独自の方法であれば問題は少ないのだが、今後の主流となるであろうと一般的に言われている方法が本当に役立つのかは導入前に検討する必要があるだろう。一例としてボリューム・ビジュアライゼーションを考えてみる。

2. 大規模計算における可視化

大規模計算における可視化は、計算に適した機種と可視化に適したものの違いから、ハードウェアの性能の向上まかせという面が強くあらわれる。現時点では10万点程度の構造格子であれば、パソコンでの可視化も可能であるし、100万点であっても、GWSで処

理ができる。数千万点となるとどうか。並列化等による手法の効率化、解法への組み込みなど考慮すべき点は多い。しかし、それ以上に大規模計算の特性を考え、それに適した方法を模索する必要がある。そのためには、大規模計算のもつ二つの面（全体から得られる大規模構造、部分部分の詳細な構造）を同時に考えなければならない。図1に球後流の圧力分布を示す。レイノルズ数は40万、計算に用いた格子数は約1200万点(300x250x161)である。図1(b)は、1(a)を拡大したもので、剥離位置は球面上の縞模様の最後の部分であることが確認されており、それより後流側で軸対称性が著しく崩されている。

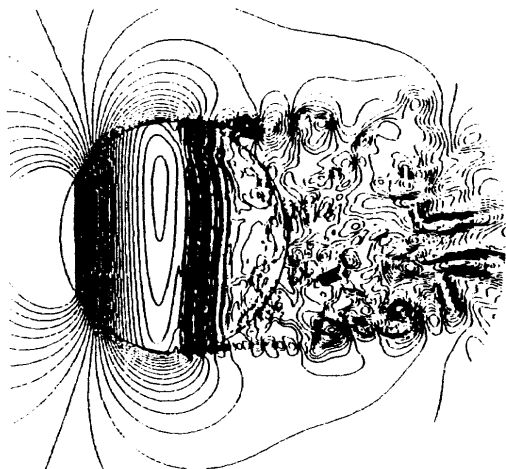


図1(a) 物体面と対称面上の圧力分布

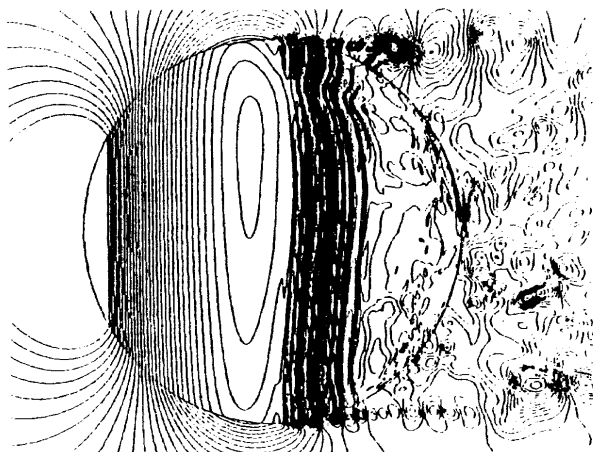


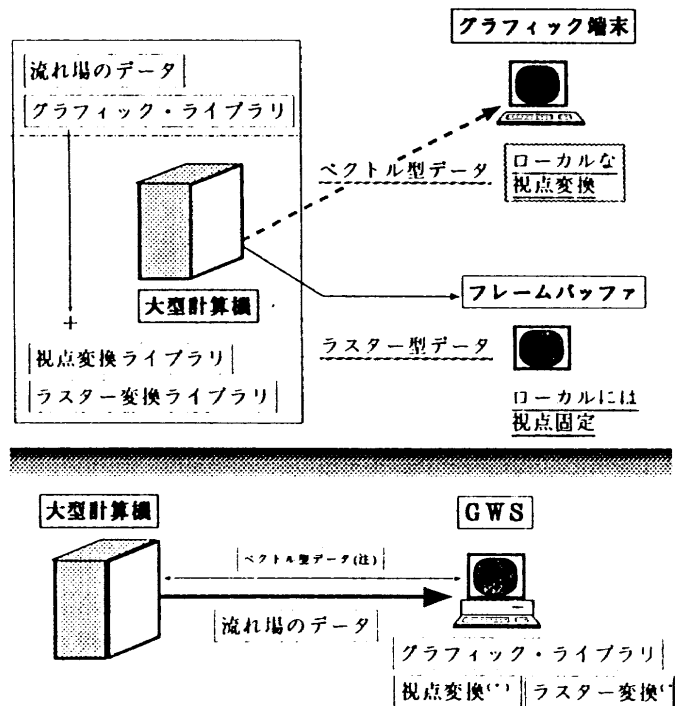
図1(b) 図1(a)の拡大図

実験によると、このレイノルズ数では抵抗の急減現象が始まっていて、計算によって詳細な様子が捉えられたかに思える。実は、この可視化は実験的事実とこれまでの経験をもとに等値線のレベルとか見方等がある程度決めて行っている。物体形状が単純なことも見るべき領域を絞り込むことに役立っている。それでも、着目すべき現象は、ある領域に集中しているものの、前もって特定することはできない。このため同じ可視化手法を用いてもいくつかの見方をする必要がある。そして、部分部分の構造は全体の構造か

ら見ると空間のコヒーレンスが良いとはいえない。

さて、大規模計算の二つの面を計算環境から考えてみよう。図2に可視化システムからみた計算環境の例を示す。全体を見る場合、ディスプレイの精度（画素の数）が中心に考えられることが多く、画面解像度が基準となる可視化の体系があらわれる。大型計算機上でイメージデータを生成後、結果はフレームバッファに送られるというシステムが考えられる。後述するように、ボリューム・ビジュアライゼーションの利点が大規模計算に適したものであるとする研究の根拠は、画面解像度が基準になる可視化が大規模計算の一つの特徴を捉えているということに起因する。部分を見る場合は、データ面でいえば粗い計算の集まりと見なせるので、次の三つのデータ選択がその時々で考えられる。

- (a) 全体から特徴的な部分が抜き出されていれば計算データそのもの
- (b) 見たい量が決定できればベクトルデータ
- (c) 見たい量と視点が決まればイメージデータ



(*)視点変換、ラスタ変換は、ソフト的に行う場合、ハード的に行う場合、グラフィックスエンジンのようにその部分だけ高速化する場合等がある。

図2 可視化からみた計算環境

球まわりの流れの可視化がどの様に行われたのかを紹介しよう。格子点の座標値に対するデータ量は144Mbytesで、流れの諸量として圧力と速度をとったので、それに対する一時刻のデータが192Mbytesであった(単精度とする)。GWSやグラフィック端末(TSS端末として接続されている)で会話処理できず、ペク

トル型のデータを大型計算機側で生成し、一時データとして保存後、視点を変更して、二次元ディスプレイで処理した。等圧線を求めるために要した計算時間は約10分、それをグラフィック・データとして転送、そして、一枚の図を描かせるのに約10分かかった。問題点として、データ転送をどうするか(ハードウェアの発展、どのようなデータ形式がよいか)、データをどう蓄えるか(データ圧縮の問題)、データ処理に要する時間を短縮できないか(可視化手法の高速化)などが取り上げられた。その後、アルゴリズムの高速化によって、等値線の計算は約1/2に、表示時間は数秒になったが、すべてが解決されたわけではない。

大規模なデータを扱う上で、「どのように特徴的な部分を見つけるか」が最も重要ではあるが、一般論を導くことは難しく、流れの研究者は事例毎に方法を変えているようである。このテーマは別の機会に話すこととし、この節の最後として、可視化法によるグラフィック・データのデータ量を見積もってみたい。三方向にほぼ同数の点である三次元データを考える。総格子点数を n とすると次のようにデータ量が見積もれる。

・ベクトル型

(a) 点や線による可視化結果から

- ・格子図、ベクトル図、等値線図

$$(n^{\frac{2}{3}}) * O(1)$$

- ・仮想粒子による可視化

$$\text{発生粒子数} * \text{積分ステップ数}$$

(b) 多角形による可視化結果から

- ・格子面図

$$(n^{\frac{2}{3}}) * O(1)$$

- ・等値面図

$$(n^{\frac{2}{3}}) * 4 * \text{枚数} * O(1)$$

・ラスタ型(イメージデータ)

$$\text{画素数} * 3 \text{bytes} \quad (\text{例}) 640 * 480 * 3 : \text{約 } 1 \text{Mbyte}$$

このように計算環境に合った可視化を行うために、既存のシステムをデータの構造から見直す必要がある。但し、解析によって得られた知見を可視化システムにフィードバックさせることで、より効率的なデータのハンドリングが可能になることを忘れてはならない。

3. ボリューム・ビジュアライゼーション

新しい可視化手法として、ボリューム・レンダリングが紹介されてから既に数年が経過した。騒がれたわりには使われていないというのが正直な感想であるが、方法自身は未だに新機軸として扱われる。その理

由としては、大規模計算に適した可視化手法としての期待とハードウェアの進歩による効率面の改善である。

最近では、ボリューム・レンダリングはボリューム・ビジュアライゼーションの中のレンダリングの操作として位置づけられるので、ここでは、手法をボリューム・ビジュアライゼーションと呼ぶことにする。著者は、ボリューム・ビジュアライゼーションを『二次元等値領域表示の三次元への拡張』と位置づけてきた[5,6]。

等値領域表示の利点は、図3に示すように等値線表示において特定領域の存在位置を明瞭にすることにある。特に、大規模計算において、はじめに特徴的な部分をさがすという作業に適している。それでも使われなかったわけは効率の悪さと定量性である。等値領域表示を等値線表示の代用とする場合、物理量の定量的な分布をも表現できることが前提となる。しかし、ボリューム・ビジュアライゼーションの場合、定量性があいまいになることが指摘されている[5]。これは可視化の精度を考える上で重要である。

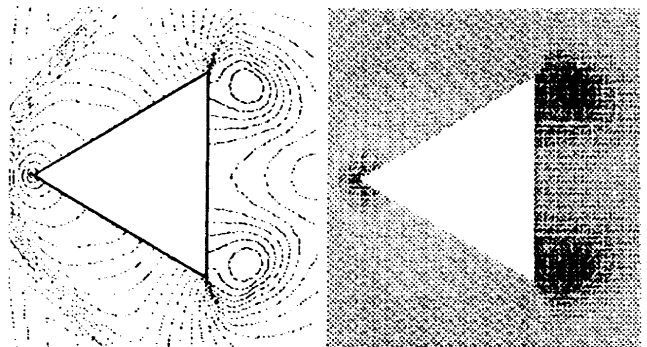


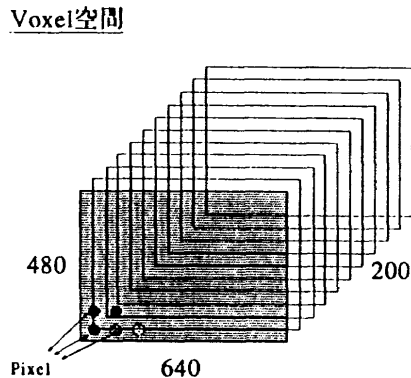
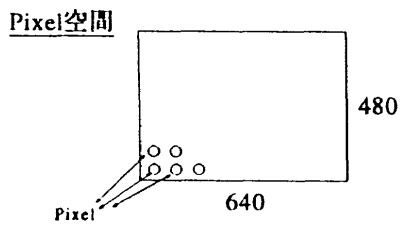
図3 等値線表示と等値領域表示

効率の悪さと定量性の問題があっても、大規模計算に適するとされたのは効率面を補う次の二つの利点が考えられたからである。一つはいわゆる比較論で従来の手法を大規模なデータの可視化に用いたとき、操作の数が増えるために効率的でなくなることである。もう一つは図4に示すように格子点数が増えると、ボクセル空間が三次元のデータ空間と一致するという考えである(この論文ではボクセル空間を均質なデータの並びとして表現する。このため、CGの文献にあるように一般の三次元データ構造の総称としてのボクセルという呼び方はしない)。後者は、ボリューム・ビジュアライゼーションにとって理想的な状況である。しかし、CFDの大規模計算に当てはまるわけではない。というのは、CFDの場合、格子が均質であることは稀で、かつ、前節で述べたように空間のコヒーレンスが悪いものが多い。このため、元のデータからボクセル空間を構築する必要が生じる。

この操作をボクセル・サンプリングと呼ぶこととする。

図5に二次元でボクセル・サンプリング(ピクセル・

サンプリング)の概念を示す。計算格子は均質なものではないので、例えば図5の×印で示したボクセル空間での値を計算格子から補間しなければならない。



計算空間とPixel空間(二次元)、Voxel空間(三次元)が一致すれば、スカラー値に対応した色をPixelに与えればよい。

図4 ボクセル空間の概念図

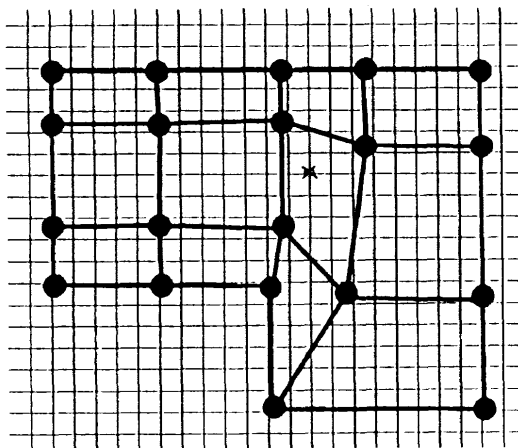


図5 ボクセル・サンプリングの概念図

この時、ある点が多角形の内か外かの判断が必要となる。この判断のための効率的なアルゴリズムは提案されているものの探索点が増えると計算時間は膨大なものとなる[2]。このため、大規模計算において計算格子点数の増加とボクセル数の増加は効率面でボリューム・ビジュアライゼーションを非現実的なものとする。

さらに、データの可視化時に切り出されたボクセル空間で視点変換を行う場合、一度のサンプリングで済むが(視点変換毎に行うレンダリングのためのサン

プリングをリサンプリングと呼ぶこととする)、計算精度は元のデータが最も良いので、可視化の精度という問題が生じる。

CFDのデータの場合、ボクセル・サンプリングとリサンプリングを同時に行い、レンダリングにより画像イメージを決定する方法が一般的となる。よって、大規模計算に対するボリューム・ビジュアライゼーションを行う場合、サンプリングの効率化が重要になる。また、精度を考える上でレンダリングの最適化も考慮しなければならない。

3.1 サンプリングの効率化

最も簡単なサンプリングの方法は先述した図5の方法である。三次元での例を図6に示す。例えば、計算格子点数を100万点、ボクセルの数を640x480x200とすると、約6000万点のボクセルが、どの6面体(約100万個)に属するのかを判断する必要がある。この場合、近傍の情報を使っても計算負荷は大きい。かなり効率的なアルゴリズムでも約6万点の格子から1280万点のボクセルを探すために、実行性能が約15MFLOPSの計算機を用いて、563秒の計算時間を要した[6]。悪いことには、単純なサンプリングの場合、ボクセルが計算領域の内か外かの判断も必要となる。計算領域より大きなボクセル空間を用いる場合のサンプリングの負荷はさらに増大する。

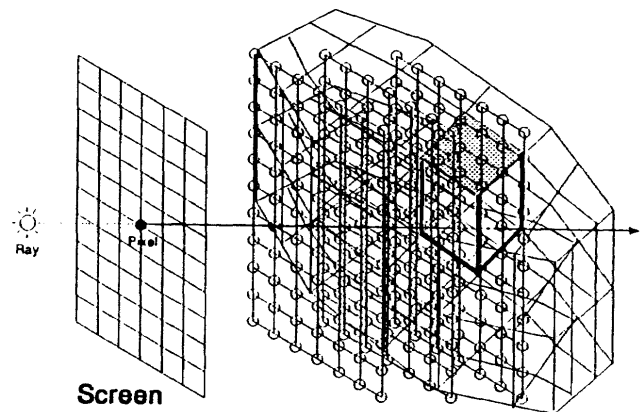


図6 三次元サンプリング(Ray Cast Algorithm)

探索の時間を短縮するために、計算格子を基準としたアルゴリズムが考えられた。計算格子を多くの多面体の集まりと見なし、3D-DDAによってボクセル空間を構築する方法である。3D-DDAは、プリミティブを中心とするため、CFDのデータのように計算格子が一様でないものに対しては可視化の精度が画像解像度と密接に関連する。画像解像度より粗い計算格子に対しては効率的で可視化精度もよい。しかし、前

もってボクセル空間を用意しておかなければならず、非常に大きなメモリを必要とする。そのため、3D-DDAの特徴を活かした別の方法が考えられた。一例は、まず計算格子を視点座標系に変換し、計算格子を視点座標の一方にスライスしながら探索する方法である(図7/1990年頃のCGの会議の参加者から聞いた話である。残念ながら参考文献等がわからない)。本稿では、その中でも、z方向にスライスされた多角形の集まりをピクセル化してサンプリングを行う方法を採用した。この方法を、z-スライス法と呼ぶ。図8に、球対称的な計算格子に波状に分布したスカラー値の空間分布を可視化した例を示す。サンプリングに要した

$\frac{1}{4} * n + \frac{3}{4} * n^{\frac{3}{2}}$ に比例し、s2に要する時間は、 $n^{\frac{3}{2}} + m^{\frac{1}{2}}$ に比例する。実際の計算に要した時間を格子点数を変えて調べた。結果を表1に示す。ボクセル数は640x400x50の1280万点、計算に使用した機種はPC98211p(i486DX2/66MHz)で、機種性能としては約6MFLOPS、実行性能として、2~3MFLOPSと予想されている。

表1: 計算時間/秒

格子点数	s1	s2	s1+s2
20x20x20(1)	19 (1)	47(1)	66 (1)
20x40x40(4)	35 (1.8)	73(1.6)	108 (1.6)
40x40x40(8)	57 (3)	110(2.3)	167 (2.5)
50x50x50(15.6)	95 (5)	149(3.2)	244 (3.7)

計算機性能を加味した比較では、z-スライス法は従来の方法に比べ約10倍速くなっている。さらに、この方法は分散計算環境に適していて、k個システムのPVMでは、s1、s2ともに、計算時間が $\frac{1}{k}$ 倍となることが期待されている。

3.2 可視化の精度

可視化の精度として考えなければならないのは、サンプリング時のエイリアス誤差とレンダリングの最適化である。前者は計算格子とボクセル空間の対応が図9のようなとき、物理量の空間コヒーレンスが悪いと非常に大きくなる(サンプリング定理)。後者はレンダリングによるイメージの違いが誤差の原因となる。

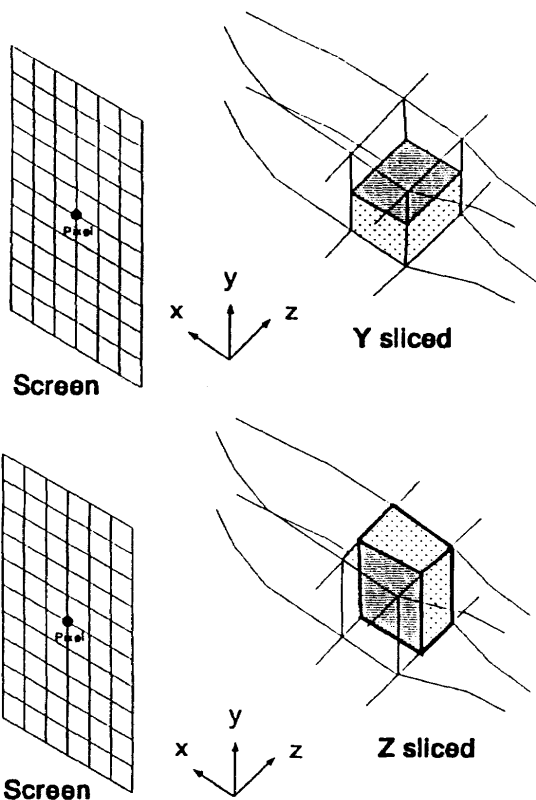


図7 スライス法

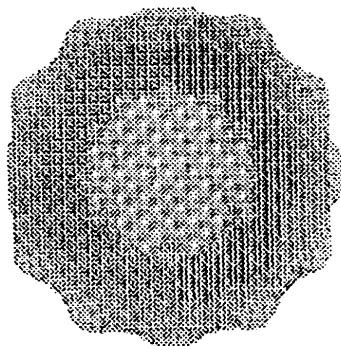


図8 モデル場のスカラーの分布

時間を、スライス処理(s1)とスライス面での探索(s2)に分けて調べてみる。計算格子点数をn、ボクセル数をmとして、粗く見積ると、s1に要する時間は、

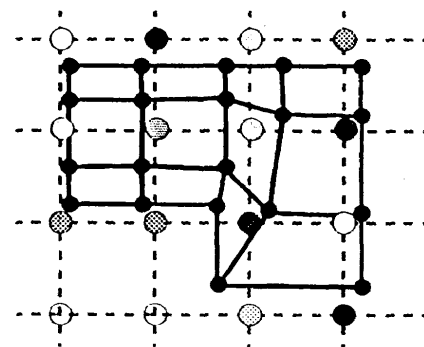


図9 Alias 誤差とサンプリング定理

レンダリングは図10のようにサンプリング点の視点からの積分によって行われる。つまり、一次元的な物理量の分布を高々数個のスカラー値(色を用いることが多い)に対応させる操作である。実験系との比較の場合、分布を積分値に対応させるための積分の方法が物理モデルの検証となり、一つの視点からの結果

で十分である場合が多い（シュリーレン法などの光学系を用いた実験や天体観測等との比較）。正確な物理モデルが得られた場合、可視化の誤差はエイリアス誤差に起因する。

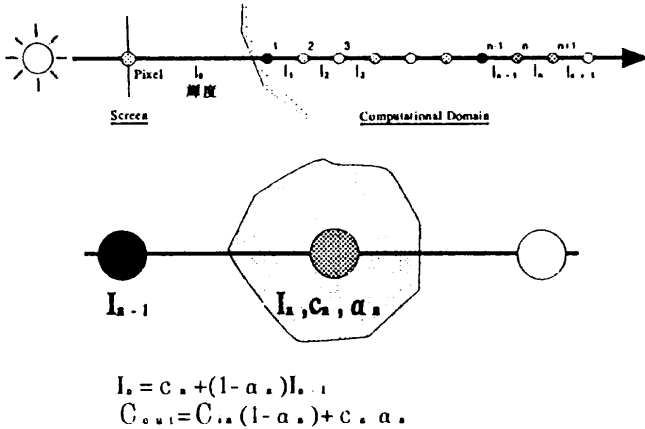


図10 レンダリングの方法

一般的なボリューム・ビジュアライゼーションの場合、物理量の分布の違いによる差異を表す必要があるが、一つの視点からの結果で表現することは難しい。図11はレンダリングの方法を変えた回転楕円体を過ぎる流れ場の動圧分布である。エイリアス誤差とレンダリングの方法による任意性がどの程度結果の解釈に影響するかは、三次元の定量的な分布が如何に正確に求められるかという問題を提起する。この問題は、ボリューム・ビジュアライゼーションの精度を吟味する上で非常に重要なものとなる。紙面の都合で紹介できないが、視点を変化させて求めた何枚かの図から三次元データを再構築することによって、元のデータと比較し、精度を見積もるといった試みを行っている。方法や結果は近々公表する予定である。

4. おわりに

画像データの観点からは、大規模計算の空間コヒーレンスは良いとは言えない。このため、全体と部分を相互に繰り返し調べる必要があり、可視化のパラメータの絞り込みが難しくなる。これは、グラフィック・データ量が一定であるというボリューム・ビジュアライゼーションの利点やはじめに特定領域を探すという目的をも打ち消してしまう。この意味でボリューム・ビジュアライゼーションの効率化と高精度化が急務となる。効率面の向上のためにアルゴリズムの並列化や専用ハードウェアの開発が行われている。高精度化をも実現する手段として、プリミティブ中心の3D-DDAが推奨されている。しかし、メモリ容量が膨大となるため、3D-DDAの特徴を備えたいくつかの方法が模索されている。その中でもスライス・アルゴリズム

という方法が適当であると思われる。精度の面では、x- または y-スライス法が良いが、本稿では、省メモリである z-スライス法を採用した。z-スライス法を用いれば、パソコン環境でも、一つのイメージを数分で生成できる。さらに、分散計算環境に適した方法であることが確かめられている。このため、分散計算環境のもとで流れの解析システムを構築した場合、可視化法の中に効率的なボリューム・ビジュアライゼーションを組み入れることができ、より強力な可視化システムの構築が可能になるものと思われる。

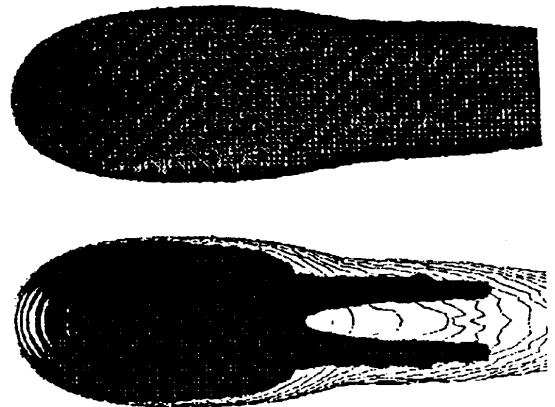


図11 動圧分布の二つの表現

参考文献

1. 白山 晋, "仮想粒子追跡法による流れの可視化", 第4回数値流体力学シンポジウム講演論文集, December, 1990, pp.483-486.
2. Shirayama, S., "Processing of computed vector fields for visualization", Journal of Computational Physics, vol.106, no.1, 1993, pp.30-41.
3. 白山 晋, "局所的な補間を用いた風上法について", 航空宇宙技術研究所特別資料 SP-19, December, 1992, pp.199-204.
4. 白山 晋, "ある種のベクトル場を可視化するための局所的補間について", 第7回数値流体力学シンポジウム講演論文集, December, 1993, pp.697-700.
5. 白山 晋, "CFDにおける可視化について", 東北大学流体科学研究所技術報告 (No.4), February, 1992, pp.13-40.
6. Shirayama, S., "Flow visualization by ray casting in curvilinear systems and feature-oriented image enhancement", Proceedings of the 1st European Computational Fluid Dynamics Conference, Brussels: Elsevier, pp.131-138, September, 1992.