

WWW 上でのデータ可視化システムの開発

上沢 和真[†] 土井 章男[†] 小山田 耕二^{††} 三中西 信治^{††}

[†]岩手大学工学部情報工学科

^{††}岩手県立大学ソフトウェア情報学部

^{††}アーレスティール研究所

概要

本論文では、WWW (World Wide Web) 上でデータ可視化に適したアプローチと我々が現在開発中の Java Visualization System (JVS) について述べる。JVS は JAVA 言語で開発された Web 上で動作するデータ可視化システムであり、様々なオペレーティングシステム上で動作する。また、JVS を CFD (Computational Fluid Dynamics) シミュレーション結果に適用し、その有効性、計算時間、そして視覚効果に関して評価を行った。

キーワード：データ可視化，WWW，JAVA，ネットワーク，コンピュータグラフィックス

Network-centric Visualization System

Kazuma KAMISAWA[†], Akio DOI[†], Koji KOYAMADA^{††}, Sinji SANNAKANISHI^{††}

[†] Department of Computer Science, Faculty of Engineering, Iwate University

^{††} Faculty of Software and Information Science, Iwate Prefectural University

^{††} Ahresty R&D

Abstract

This paper describes an approach suitable for data visualization on world wide web (WWW) and Java Visualization System (JVS) that we have been developed. JVS is working as an applet on WWW browser, and running on a variety of Operating Systems. We also apply JVS to the results of computational fluid dynamics (CFD) applications, and evaluate it in terms of the effectiveness, the computational time, and the visual effects.

Keywords: Data Visualization, WWW, JAVA, Network, Computer Graphics

1 はじめに

現状の可視化システムを大別すると、シミュレーションと可視化を同一の計算機上で行うスタンドアローン型と X-Window System 等を利用したクライアント・サーバ方式に分けられる。前者はユーザがその計算機の側にいなければならないが、後者はネットワーク上のサーバ側で可視化処理を行い、クライアント側で画像表示するため、その制約はない。

しかしながら、X-Window System 等を用いたクライアント・サーバ方式では常時ネットワーク間で接続しているため、ネットワークに負荷がかかっている場合は応答が非常に悪くなる。またネットワーク間の距離が遠い場合やインターネット上では実用的でない。さらに、一般の可視化ツールでは計算機ごとに独自のバイナリファイルを用意する必要があるため、システムの更新などでシステム管理者にかなりの負担をかけている。

近年、ネットワークの発達と共にネットワークを通してシミュレーション結果を可視化する要望が多い。また、システムの維持管理に対する簡略化も望まれている。我々はこれらの問題を解決することを目的に、ネットワーク上にあるシミュレーション結果を取得して、ユーザー側（クライアント側）で可視化処理を行うツールである JVS (Java Visualizatoin System) を開発している。JVS の特色は JAVA 言語で書かれており、JAVA 言語の実行形態の一つであるアプレット版としても稼動するため、Internet Explorer 等のブラウザから実行可能である。そのため、ユーザはインストール等の手間をかけずにデータの可視化を行え、最新のシステムを WWW 上から取得できる。

本論文では、JVS のシステムコンセプトやその機能を紹介し、一般に広く使われている WWW ブラウザ上の可視化ツールである VRML と比較検討する。

また、実際の CFD (Computational Fluid Dynamics) シミュレーション結果に JVS を適用して、その有効性、計算時間、そして視覚効果の評価を行う。

2 WWW 上でのデータ可視化

クライアントサーバ方式において、データ可視化処理をどの程度の割合でクライアント側が行うかにより、2つの方式が考えられる。一つはサーバ側で計算結果の可視化処理を行い、クライアント側に画像等を転送する方式である。2つめはクライアント側がシミュレーション結果をネットワークを通じて取得し、可視化処理もクライアント側で行う方式である。

前者の代表的な例はシミュレーション結果を等値面や流線等のジオメトリデータ (VRML ファイル) に変換し、クライアント側に転送して VRML ビュワーを用いて可視化を行う方式である。VRML (Virtual Reality Modeling Language) はインターネット上で3次元シーンを記述でき、インタラクティブ性を持つ言語である。VRML ファイルの表示には WWW ブラウザ上にプラグインの形で提供されているビューワーを用いて行う。

VRML を使用して可視化を行う場合、以下の手順となる。

1. サーバ側でシミュレーション結果を VRML ファイルに変換する
2. 変換された VRML ファイルをクライアントである WWW ブラウザに送信する
3. 送信された VRML ファイルはプラグインにより可視化される

本方式の場合、シミュレーション結果を VRML に変換するプログラムの開発が必要であり、この変換プログラムはサーバ側で実行される。クライアント側への転送のプログラムは基本的に必要ないが、クライアント側から VRML ファイル作成指令を受け取るプログラムは必要である。

この方式の利点はクライアント側で仮想空間内に用意されたシミュレーション結果を視点等を自由に変化させながら可視化できる点である。問題点はユーザ側では VRML の表示のみを行うため、異なる VRML ファイルが必要となった場合、サーバから再度 VRML ファイルの作成および転送を行わなくてはならない点である。

例えば、等値面の定数を変更する度に VRML ファイルを作り直す必要がある。

それに比べて、JAVA 言語を用いて可視化システムを構築する場合、クライアント側ですべての可視化処理を行える。

JAVA 言語は C++ に似た文法を持ち、分散処理やネットワークに対応したライブラリを持つ言語である。JAVA 言語には2つの実行形態（アプリケーションとアプレット）があり、アプリケーションは単独で JAVA 仮想コンピュータ（JAVA VM）上で実行できる。アプレットは WWW ブラウザに内蔵されている JAVA VM によって実行される。

3 JVS (Java Visualization System)

3.1 JVS 概略

JVS (Java Visualization System) は JAVA 言語によって書かれており、JAVA コンパイラでバイトコードにコンパイルされた後、JAVA VM (JAVA 仮想コンピュータ) 上で実行される。JVS のアプレット版は機能的にアプリケーション版と同じであるが、Internet Explorer や Netscape Navigator 等のブラウザソフトから起動できる。そのため、システムの存在するアドレスさえわかれば、面倒なインストール手順を行う必要もなく、最新のシステムを WWW 上から取得できる。

JVS は起動後、ユーザからのレンダリング要求によってシミュレーション結果を保存しているサーバにアクセスし、データを取得し、可視化処理を行う。一度、シミュレーションデータを取得すれば、視点やレンダリングパラメータを変更してもサーバ側にはアクセスしない (図 1)。

JVS の開発環境には、JDK 1.1.3[5] を使用し、実行環境として Windows NT 4.0 (Pentium Pro 200Mhz 128Mb メモリ) 上で JAVA インタプリタを搭載した WWW ブラウザ (Internet Explorer 4.0) と同 OS 上で JDK 付属の JAVA VM を用いている。

3.2 JVS 可視化機能

JVS で扱えるデータはスカラ場とベクタ場であり、対象とするシミュレーションの格子構造は、正方格子、構造格子、非構造格子である。JVS は4つのオブジェクトからなり、ボリューム、ジオメトリ、イメージ、ユーザインタフェース (GUI) から構成される。ボリュームオブジェクトはシミュレーション計算結果をネットワーク上の計算サーバから読み取り、ボリュームデータに変換する。ジオメトリオブジェクトは作成されたボリュームデータから流線、等値面、断面のジオメトリデータを作成する。そしてイメージオブジェクトは作成されたジオメトリデータをレンダリングして画像表示を行う。等値面の定数や流線発生場所のパラメータが変更されると再度ジオメトリデータを作成し画像表示する (図 2)。

JVS の GUI は、JAVA 言語のライブラリに用意されている AWT (Abstract Window Toolkit) を用いて作成している。GUI は画像を表示する Canvas とパラメータ入力用のテキストフィールド、レンダリングを始めるボタンからなる。

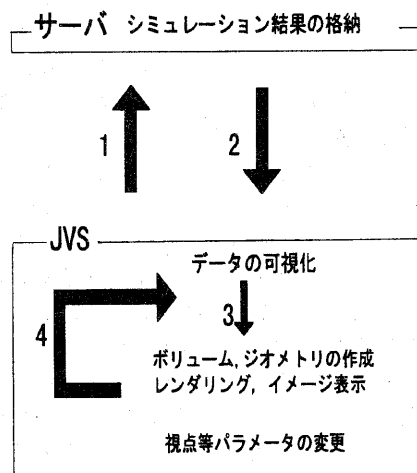


図 1: システムの概略
System Overview

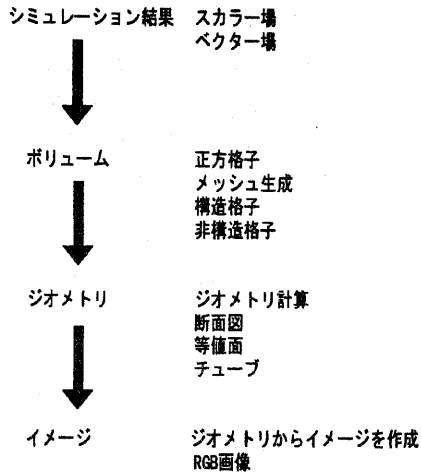


図 2: プロセスの流れ
Process Flow

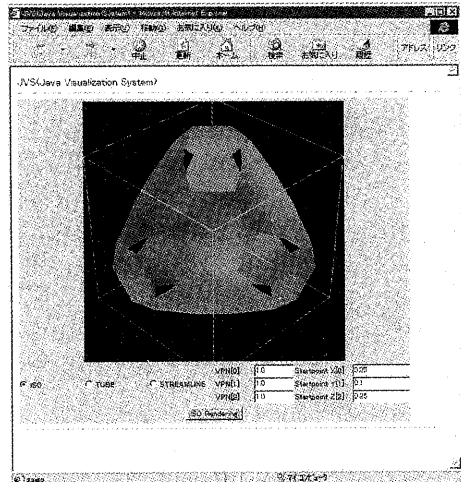


図 3: JVS による可視化例 (1)
Visualization Using JVS (1)

4 CFD シミュレーション結果への適用

4.1 JVS による可視化

図 3 は JVS アプレット版によるスカラ場の等値面表示である。アプレットのサイズは 260Kbyte である。アプレットがクライアントに転送され、アプレットが動作するまでには約 30 数秒かかった。等値面のポリゴン数は 80 個でレンダリング開始ボタンを押してから数秒で処理は終了し、画像が表示された。この場合、JVS はレンダリング開始ボタンが押されると、シミュレーション計算を行ったサーバから離れた場所でネットワークを通して自動でデータの取得とその可視化を行った。図 4 は等値面表示でポリゴン数 840 個で処理に 10 秒ほどかかった。図 5 は JVS アプリケーション版による等値面表示である。アプリケーション版はコマンドライン上でシミュレーション結果が置かれているサーバの URL を入力することでデータ取得を行い、可視化処理を行う。生成された等値面のポリゴン数は 79507 個で約 4 分で表示された。図 6 もアプリケーション版による速度場から流線をチューブ表示したものである。各図の格子構造はすべて正方格子である。

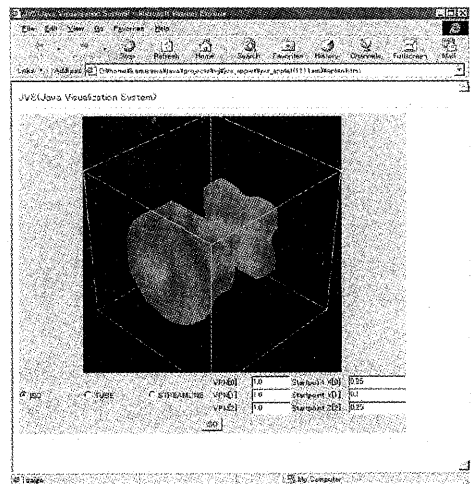


図 4: JVS による可視化例 (2)
Visualization Using JVS (2)

4.2 VRMLによる可視化

VRMLによる可視化は断面や等値面等のジオメトリデータをVRMLフォーマットに書き出し、イベント処理等はVRMLファイルを直接編集することで行なえる。図7と図8はスカラ場の可視化例であり、アニメーションの開始と終了時の画像を示したものである。この例では開始から終了まで、あらかじめ指定された間隔で断面図をテクスチャマッピングにより作成しておき、順次表示対象を変更している[2]。また、視点等もVRMLの世界の中で自由に変わることが可能である。表示速度はファイルのダウンロードが終了してから約10秒である。

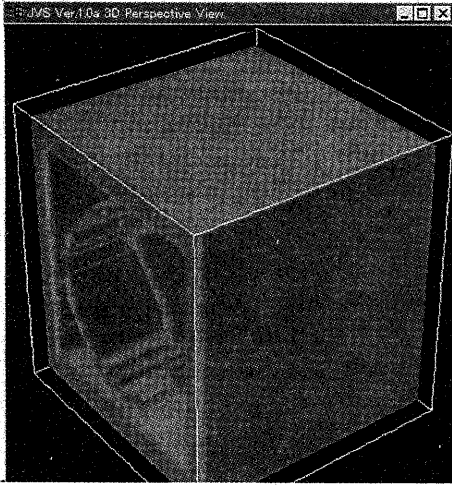


図 5: JVS による可視化例 (3)
Visualization Using JVS (3)

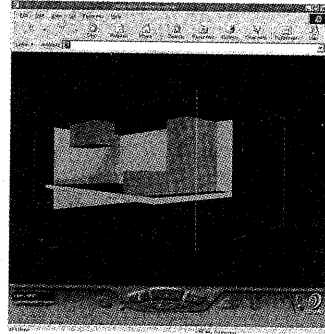


図 7: VRML による可視化例 (1)
Visualization Using VRML (1)

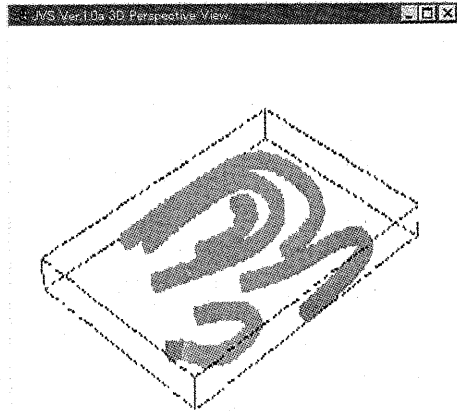


図 6: JVS による可視化例 (4)
Visualization Using JVS (4)

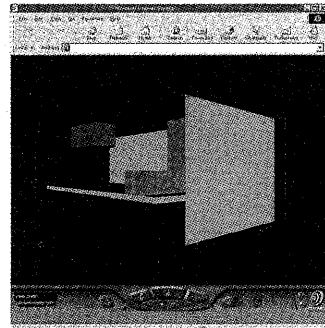


図 8: VRML による可視化例 (2)
Visualization Using VRML (2)

4.3 システム評価

VRML 言語を使用する場合は、サーバ側で VRML ファイルを作成し、クライアントである VRML ビューワに送信して可視化する。これはネットワークに負荷がかかっている場合でもサーバ側から VRML ファイルの転送を終了すれば、視点変更等のユーザ側の処理にはなんら影響を受けない。そのため、多数のユーザにシミュレーション結果の表示を見せたい場合には有効である。また、VRML をサポートする多くの表示ツールが利用可能であるため、全体のシステム開発コストも小さい。しかしながら、VRML ファイルの再構成が何度も必要な場合には、サーバ側から計算後の VRML を取得する回数が増えるため効率が悪い。

一方、JVS はサーバ側からシミュレーション結果を受け取るとクライアント側ですべての計算を行うため、サーバ側との通信はデータ取得時のみである。そのため、サーバ側で処理を行う VRML 方式に対し、クライアント側で処理を行う JVS はネットワークへの負荷を抑えられる。これは、ユーザが何度も可視化処理を行う必要がある場合に有効である。また、CPU や OS に依存しないシステムを開発できることも長所の一つであり、システム構築の自由度は大きいが開発コストは大きくなる。

計算機を選ばないクロスプラットフォームという点では VRML は機種ごとにビューワが用意されており、JAVA 言語では JAVA VM が用意されている WWW ブラウザであれば実行できる。そのため、両方式もネットワークに接続していれば（もしくはデータが手元にあれば）、共に機種や OS を選ばずにデータの可視化することが可能である。

5 おわりに

本論文では、WWW 上でのデータ可視化システムである JVS の概要と機能について述べ、実際の CFD シミュレーション結果に応用し、その有効性、計算時間、そして視覚効果に関して、評価を行った。JVS の今後の開発課題としては、以下の3点が上げられる。

1. パフォーマンスの向上

2. ユーザインタフェースの改善

3. VRML と JAVA を用いた可視化環境の構築

JVS はクロスプラットフォームな可視化ツールを目指しているが、描写速度等、さらに改善を必要とする場合には JIT(Just In TimeCompile) の使用による速度向上が考えられる。また、ネットワーク上から転送されるバイトコードの大きさを小さくし、ネットワーク上を高速に転送できるようにする必要がある。

機能的には、現バージョンの JAVA 言語が OpenGL のような汎用 3D ライブラリを持たないため、VRML のようにユーザが自由に視点等を変更できるインタフェースの作成は容易でないが、今後 JAVA3D 等の利用が考えられる。また、VRML と JAVA 言語の得意分野を利用した可視化環境の構築技術も今後重要になってくる。そのためには、サーバ側とクライアント側の機能分担を考え、シミュレーション結果や VRML ファイルを保存する可視化サーバの設計が必要である。

6 謝辞

本研究は、(財)大川奨学金により補助を受けた。ここの感謝の意を示します。

参考文献

- [1] Rodger Lea, Kouichi Matsuda, Ken Miyashita, "JAVA for 3D and VRML Worlds", プレンティスホール出版, 1997.
- [2] Koji Koyamada, "VRML による可視化のためのデータ削減技術", 流体力学学会誌 "ながれ", pp323-327, 1997.
- [3] Koji Kawaguchi, "Java アプリケーションから見る Java の可能性", JAVA PRESS1, pp. 20-38, 1998.
- [4] Yoshiyuki Suzuki, "100% Pure Java の留意点と問題解決方法", JAVA PRESS2, pp. 2-8, 1998.
- [5] Sun Micro Systems Inc. , "JDK1. 1. 1Docum entation", 1997.