

# Web 上の 3 次元物体の形状表現

## ～ Java 3D, XVL の表現方法 ～

2000 MT022 市橋 翔輔 2000 MT083 櫻井 哲平

指導教員 青山 幹雄

### 1. はじめに

本研究は、Web 上の 3 次元表現フォーマットである Java 3D, XVL を用いて、3 次元物体を定義し、その表現方法について評価した。物体を定義する上で扱う形状データとポリゴンメッシュを中心に、どのような構造で物体が生成され、どのようにすればデータの情報を軽量化できるかを検討した。

### 2. Web 上での 3 次元表現方法

Web 上での 3 次元表現方法には、VRML, XVL, Java3D がある。VRML (Virtual Reality Modeling Language) は、Web 上の 3 次元表現の基本的な方法である。基本的には、ポリゴンという三角形や四角形の平面の集まりによって形状を表現している。このフォーマットもまた、多くの CAD や CG ソフトから出力する事が可能である。ポリゴンで形状を近似して表現しているため、精度を要求される製造分野での利用は難しく、また、見る事に特化した e-Commerce 等の利用を想定して、詳細なモデルを表現しようとすると、大量のポリゴンが必要となり、データ量が膨大になる。このため、VRML の改良が提案されている。本研究では、この XML をベースにした XVL と Java をベースにした Java3D を取り上げて、研究した。

### 3. XVL

#### 3.1 XVL とは

XVL (eXtensible Virtual world description Language) は、VRML の後継である。X3D (eXtensible 3D) 規格の一つである。VRML は、VRML1.0, VRML2.0, VRML97 と発展してきた。X3D 技術を拡張した表現言語が、XVL 技術である。従来より、けた違いに軽いデータサイズで形状を表現し、インターネット環境でデータをやりとりするためのフォーマットである。XVL は、 $10^1 \sim 10^3$  mm という高い精度で CAD データを 1/100 以下に軽量化することが可能である。これにより今までスタンドアロンの環境に蓄積されてきたレガシーデータをインターネットで利用することが可能になる。また、ラティス構造という独自の格子の情報によって制御され、これにより従来のシステムでは実現できなかったレベルでの直感的な形状操作とデータ軽量化が実現される。

他の 3D 表現形式では、実現できなかった高い品質(精度)のデータを少ないデータ量(軽量)で送ることを実現した。また、ポリゴン表現に加え、曲面データ表現をサポートすることによって、高精度を実現し、ネットワークへの負荷を最小にして、従来の VRML のようなポリゴン転送を行わず、ネットワーク上は軽量のラティス構造(格子の情報と重みなどの付加情報)や曲面構造を転送する。

高速転送された XVL を受け取った PC 上で表示用のポリゴンデータを指定された精度で生成し、高速 CPU でリアルなイメージを再現することによって、他の 3D 表現形式では実現できなかった高い品質(精度)のデータを少ないデータ量(軽量)で送ることを実現した。

図 1 で示すのは、既存の 3D データを XVL データに変換することによって、既存の 3D データを 1/100 以下に軽量化することが可能になる。その軽量化した XVL データでインターネット回線を介して、他のホストに送信し、XVL データを高速 CPU で処理することによって、既存の 3D データを参照することができる。また、情報の構造を規定することができ、3D 構造のような複雑な構造でも定義することが可能な XML を用いることによって、ネットワークとの親和性の問題を解決した。

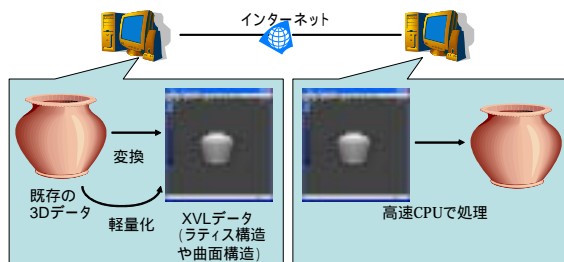


図 1 XVL の表現の仕組み

#### 3.2 XVL での物体生成について

3 次元形状を自由に作成・変形する為のツール XVL Designer を用いた。立方体・球・円錐・円柱の基本形状を用いて、それらを自由な発想で変形して、思いどおりの 3 次元形状を完成させる事が出来る。図 2 に壺を生成する過程の図を用いて示す。

まず、基本形状を 4 つの選択肢から立方体を選び生成し、拡大する。その後、丸み形状を ON にし、(a) で壺に丸みを

出し、面を押し出す。(b)で面を押し出し、(c)で壺の太さを拡大し、調整し、首の長さを調整する。以上の動作を基本動作として行う事により、壺が作成される。

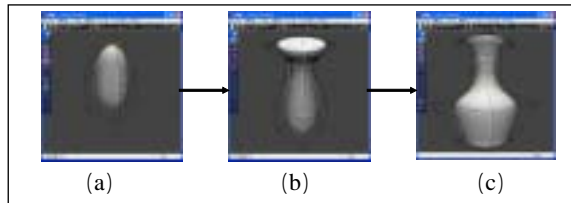


図2 三次元物体を生成する過程

### 3.3 研究課題

図3は、現XVLアプリと新XVLアプリの比較を示す。XVLのアプリケーションが追い求めているのは、現アプリケーションに比べて、新アプリケーションで読み込み速度とメモリ消費をいかに良くするかである。読み込み速度を速くしたり、メモリ消費を少なくする問題は、3次元設計で扱われるデータを軽減させる必要がある。その為、形状データを幾何学データで受け渡すことで軽量化できるのではないかと課題が出てくる。

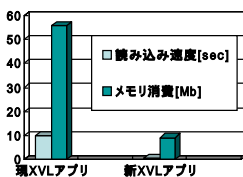


図3 XVLアプリの比較

## 4. Java 3Dの概要

### 4.1 Java 3Dとは

Java3Dとは、Java2で3次元グラフィックスを実現するための公式なオプションパッケージAPIである。1998年に、Sun Microsystems, Inc. よりリリースされた。Java3Dは、Java2でマルチメディアを扱うための標準的なAPIであるJava Media APIのひとつに分類されており、3次元音響も扱う。それまで、Javaで3次元グラフィックスを実現するには、計算・描写を行うコードやOpenGLなどを呼び出すネイティブコードを自分で書いたり、利用したりしなければならなかったが、Java3Dがリリースされたことで、手軽に3次元グラフィックスを扱えるようになった。

Java3Dの最大の特徴は、「シーングラフ」の考え方に基づいて設計されていることである。シーングラフは、3次元空間の物体や光源などの描写に必要なすべての情報を、一つの本構造を構築することで、3次元空間が描写される仕組みである。Java言語のオブジェクト指向と相性がよく、

シーングラフを動的にかえれば、グラフィックスも自動的に再描写でき、ユーザインタフェースに大きな自由度と拡張性を持たせた設計になっており、ヘッドマウントディスプレイで左右の目それぞれに視差のある画像を描画して見せたり、特殊な入力デバイスを使用することもできる。

### 4.2 物体の定義

Java 3Dでは、下位の3次元グラフィックスAPIであるOpenGLやDirectXと同様に、物体はポリゴンと呼ばれる多角形の集まりで構成されている。ポリゴンには表裏があり、特別な指定をしない限り裏側から見ることはできない。ポリゴンは多角形であるので、頂点の集まりで形を決めることができる。Java 3Dではポリゴンとして点・線・三角形・四角形が用意されていて、そのポリゴンの集まりで構成された形状をGeometryクラスというクラスで表現している。

描写される物体は必ずShape3Dクラスというクラスで表し、Shape3DオブジェクトはAppearanceオブジェクトと複数のGeometryオブジェクトから構成されている。Appearanceクラスは物体の見かけを表し、Geometryクラスは物体の形状を表す。Java 3Dでは、もともと球体や三角錐などを構成するクラスが用意されているが、これらもShape3Dクラスから形成されているので、Shape3Dクラスを使いこなすことができれば、これらのクラスは必要ではなくなる。しかしShape3Dは、自由度の高い反面、定義などが必要なので手間がかかる。何を描写したいかによって使いわけるのが良い。

### 4.3 Geometryクラスとポリゴン

Geometryクラスは表1のようなサブクラスによって構成されている。

表1 Geometryクラスの種類の一部

クラス	定義する形状
TriangleArray	三角形のポリゴンで構成される多面体
QuadArray	四角形のポリゴンで構成される多面体
LineArray	線分の集まり
PointArray	点の集まり
Text3D	立体化した文字列

これらのクラスでは、頂点の配列は添え字で順番に区切ってポリゴンにするのではなく、ポリゴンを構成する頂点をインデックス値で指定する。インデックス値も座標や色などの要素ごとに配列で管理されていて、コンストラクタで指定するindexCount個の配列を内部に持っている。つまり頂点座標であれば、頂点座標のインデックス値としてsetCoordinateIndex(int index, int CoordinateIndex)メソッドが用意されていて、インデックス値の配列の添え字番号をindexに、インデックス値をcoordinateIndexに設定する。

そして、インデックス値の配列が添え字番号でポリゴンの

頂点数ごとに区切られて、それぞれがポリゴンとみなされる。ポリゴンを構成するインデックス値が頂点の情報の配列の添え字番号を指し、その頂点がポリゴンを構成することになる。IndexedGeometryArray クラスのサブクラスを使う利点は、頂点の数を減らせることである。Java 3D は、座標変換や光の当たり具合による色の算出を頂点ごとに行うため、頂点の数が減れば、その分だけ効率よく計算ができる。ほとんどの形状は頂点を共有したポリゴンから成り立つため IndexedGeometryArray クラスを使うことで、それまで別々の計算していた同一の頂点を一度に計算することができる。

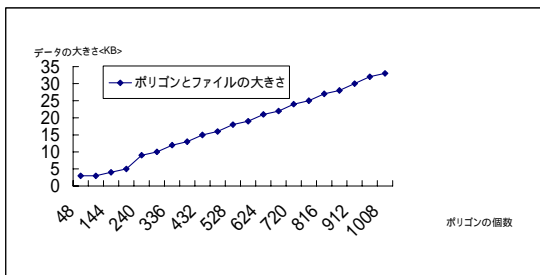


図4 ポリゴンとファイルの大きさ

図4からポリゴンとファイルの大きさはほぼ比例関係にあるのがわかる。図5は実際にポリゴンを使用し、作成したアプリレットである。



図5 ポリゴンメッシュで生成した物体のアプリレット

## 5. 三次元モデルのデータ型

### 5.1 三次元フォーマットの使われ方と課題

- ・ 協調設計, 同時設計
- ・ マニファクチャリング, ライフサイクル管理
- ・ 設計の部品化, 再利用
- ・ 設計の文書, 部品カタログ
- ・ エンジニアリングサービス, トレーニング, 教育

それらの使い方から以下のような課題がある。

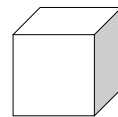
- (1) 多種情報の融合
- (2) 通信, 遠隔共有
- (3) 配布, 流通

この課題からデータの受け渡しが重要なことがわかる。3次元フォーマットのデータのことをマルチメディアデータと呼ばれ、メディアデータ型の対象となるのはテキスト、音声、画像、動画像、アニメーション(時間変化データ)などである。

### 5.2 二次元モデルと三次元モデルの情報量の比較

設計対象となる3次元立体の形状に関するデータをコンピュータ内に記憶させたものを形状モデルという。形状モデルを利用することによって、設計対象の製品としての完成後の形があらかじめコンピュータ上で検討でき構造解析を行うことができる。形状モデルを取り扱うことは設計・開発のあらゆる過程において必要とされる中核技術の一つである。

思った通りの形状のモデルをコンピュータ内に作り上げていくためには、それぞれの特徴を生かして物体を定義していくのが良い。物体を定義する際に必要なこと、注意すべきこと、クラスは何を使えばよいか、物体をリアルに見せるにはどうしたらよいかなど注意しなければいけない。形状モデルの簡単な例を図6に示す。また、2次元モデルのものと3次元モデルでは情報量は大きく異なる。物体を生成する形状モデルの他にも、アニメーションなどの時間的変化を加えるための数値パラメータや音声のような非固体のメディア型などで情報量は大きくなる。



頂点	x	y	z
0	0	0	1
1	1	0	1
2	1	1	1
3	0	1	1
4	0	0	0
5	1	0	0
6	1	1	0
7	0	1	0

図6 立方体の形状モデルと各点の座標

図7のように時間変化のデータ、非固体識別性(色、音声など)、次元数により情報量が変化する。

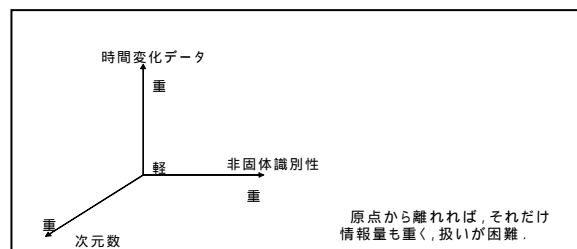


図7 情報量の大きさの関係

## 6. Java 3D と XVL の比較

### 6.1 描写モードの比較

Java 3D と XVL の描写方法を比較するため、同じ物体

(立方体を生成した(図8) . Java 3DとXVLは,それぞれ表2のように描写方法が異なる .

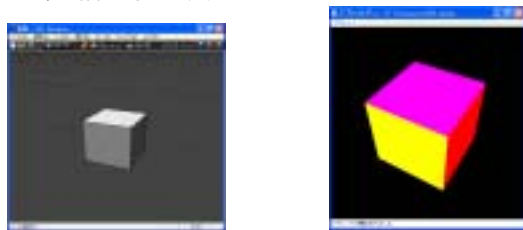


図8 XVL(左)とJava 3D(右)で作成した立方体

表2 Java 3DとXVLの描写方法の比較

3D表現方法	Java 3D	XVL
描写モード	リテインドモード	イミディエイトモード
描写対象データ	シーングラフ	幾何学データ
描写体タイミング	ライブ状態で自動描写	設計時に描写

Java 3Dは,描きたいデータをシーングラフとして構築し,そのデータを元に描写し,シーングラフがライブ状態になったときに描写されるリテインドモードである .それとは異なりXVLは,シーングラフを構築するのではなく,幾何学データ(色,頂点座標)を元にポリゴン一つずつ定義し描写していくイミディエイトモードである .

## 6. ファイルの圧縮

### 6.1 圧縮と圧縮ソフトの必要性

企業や家庭にインターネットが普及して,Web上での3次元データの圧縮・解凍ソフトの重要性は増している .例えば,電子メールにファイルを添付する際や,サイズ(容量)の大きいファイルをアップロードする際には,ファイルの圧縮を行い,ファイルのサイズを可能な限り小さくしておく必要がある .ファイルが大きくなれば,ファイルの送信時間も大きくなる .逆にファイルのサイズが小さければ小さいほど通信コストと送受信にかかる時間を軽減させることができる .実際に,世の中にある圧縮プログラムの圧縮率の比較を行った(表3) .

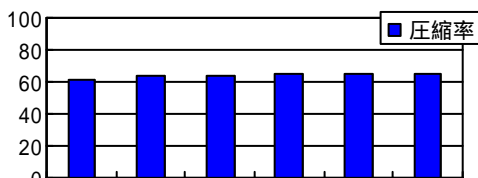


図9 圧縮率の比較

### 6.2 圧縮ツールのテスト

実際にJavaを使用し,カレントディレクトリ内のファイルをZipファイルに変換するツールを作成し,調査した圧縮率と同様に近似するかをテストした .結果を表3に示す .

表3 コマンドプロンプトからのファイル圧縮

ファイル名	圧縮前のファイルの大きさ	圧縮後のファイルの大きさ	圧縮率
Sample.txt	1228 KB	486 KB	40%
yosi.txt	441KB	344 KB	78%
Sotsuron.txt	1143 KB	712 KB	62%

## 7. まとめ

### 7.1 Java 3DとXVLを比較した結果

Java 3DとXVLとでは,同じ物体を定義し作成しても,データの大きさが大きく異なる場合もある .どちらが優れているというわけではなく,データの大きさがいけば簡単な物体であればJava 3Dの方が少ない .しかし,Java 3Dで複雑な物体を定義するのは安易でないため,XVLの方が複雑な物体の描写がしやすいとも考えられる .

### 7.2 データの大きさとポリゴンの関係

Java 3Dでポリゴンによる物体の定義を行い,データの大きさとポリゴンの数の関係を調べ, $Y=30.5X+k$ (Y:データの大きさ,X:ポリゴンの数,k:ポリゴン以外のデータ)という近似的式を出した .これにより,多数のポリゴンから構成される物体のデータの大きさが,ある程度予想することができる .

### 7.3 圧縮について調べた結果

複数の圧縮プログラムで,ファイル形式の違いを圧縮し,圧縮率を調べ,圧縮プログラムによって向き不向きのもがあり,圧縮率が大きく異なることを確認した .また,Javaを使用しZipによるコマンドプロンプトからの圧縮を行い,一般のZip圧縮とほぼ同等の圧縮を行うことができた .

## 参考文献

- [1] えんどうやすゆき,他,Java 3Dプログラミングバイブル,ナツメ社(2003).
- [2] 太田篤史,やさしくわかるJava 3D,技術評論社(2003).
- [3] 横井与次郎,JAVA&XMLサーバサイドプログラミング,秀和システム(2002).
- [4] ジョゼフ・オニール,独習JAVA,翔泳社(1999).
- [5] ロジャー・リー,JAVA+VRML JavaとVRML 2.0で作るインタラクティブ3Dワールド,プレントイスホール出版(1997).
- [6] 島谷浩志,XVL ネットワーク3D規格,CQ出版社(2002).