島根大学教育学部紀要（教育科学）第32巻　79頁～99頁　平成10年12月

Virtual Reality Modeling LanguageとJava言語によるバーチャルリアリティを用いた教材開発

川口高明*

Takaaki KAWAGUCHI

Teaching materials using virtual reality by VRML and Java

［キーワード：バーチャルリアリティ，VRML，Java，インターネット，情報教育，科学技術教育］

1．はじめに

現在，初等から高等教育にわたる教育現場では，インターネットに代表される情報通信技術を利用した情報処理（情報基礎）教育の在り方が盛んに議論されている。また実際に，すでに多くの機関で様々な試みもなされている。このような現状を鑑み，我々はインターネット用ブラウザ上にバーチャルリアリティとしての仮想3次元空間を構築する技術に着目し，情報教育および科学技術教育のための教材開発を試みている[1)]。このバーチャルリアリティは，Virtual Reality Modeling Language (VRML)[2)]と呼ばれる計算機言語で記述して設計・作成することができる。このVRMLとHyper Text Markup Language(HTML)を組み合わせることにより，World Wide Web(WWW)上のホームページにバーチャルリアリティを表現することが，簡単な方法で可能になる。そのため，このバーチャルリアリティ技術は情報教育を含むさまざまな分野において注目を集めてゆくものと思われる。その具体的方法は，VRMLの仕様に従って物体を記述したテキスト形式のファイルを，VRML用のブラウザまたはVRMLプラグインを組み込んだWWWブラウザで読み込むだけであり，これでブラウザに物体の形状が仮想3次元空間内に表示される。その後は，マウス操作だけで仮想空間内の任意の位置から任意の方向を見ることが可能になる。さらにVRML(特にバージョン2.0の仕様)によるバーチャルリアリティにおける興味深い特徴の一つは，VRMLで記述された物体を仮想3次元空間の中で運動させられるということである[2)]。その物体の運動は，インターネットに関連した計算機言語であるJavaもしくはJavaScript等で記述されたプログラムで制御される。

本研究では，VRMLやJavaというインターネット関連の計算機言語と，インターネットブラウザを利用したバーチャルリアリティ技術を用いた教材を2種類作成する。1つは，学部・研究室等紹介から科学技術教育用シミュレーション学習までを一貫して行う教材で，2つめは，バーチャルリアリティを3次元可視化の方法として用いた科学技術教育用シミュレーション教材である。以下にそれらの作成例を示し，その作成方法を説明しながら，バーチャルリアリティを用いたことによる特徴等について議論を行う。

2．学部・研究室紹介およびシミュレーション学習のためのバーチャルリアリティ教材

この章では，バーチャルリアリティと科学技術シミュレーションを組み合わせた教材についての説明と議論を行う。その作成手順として，まず最初に学部の建物の全体をバーチャルリアリティ空間に構築する。そして，建物内部構造を構築し，それから個々の研究室へとバーチャルリアリティ空間を階層的に構成してゆく。最終的には，その研究室の中でシミュレーション教材が実行可能なように設定する。ここで，シミュレーションまでの段階で，建物内部の部屋案内も行えるが，ここで仮想空間内部の任意の位置から任意の方向を見ることが可能であるというVRMLによるバーチャルリアリティの特徴を活用する。なお，本章で用いたVRMLはバージョン1.0の仕様に従い，それによるバーチャルリアリティの表示には，プラグインとしてMicrosoft VRML2.0 Viewerを組み込

＊島根大学　教育学部

んだMicrosoft製のInternet Explorer4.0を用いた。

図1に，島根大学の教育学部棟の全体のバーチャルリアリティを表示する。この中で，建物や周囲の細部を若干省略している。窓や玄関ドア等については，デジタルカメラで実際に撮影してJPEG形式の画像ファイルという形で，バーチャルリアリティ空間内に直接表示している。このように立体感の必要な建物の外郭はVRMLで構成しているが，部分的には撮影した画像ファイルを利用して，建物の細部の簡略化をしている。ただ，現実感を出すためには，建物全体を任意の方向から見ても矛盾しないように，構造的に整合性を持って構築しておかなければならない。次に，図2のように，視点を正面玄関前に移動してみる。ここで，VRMLの仕様である他のURL(Uniform Resource Locator)へのリンク機能を用いることで，この正面玄関のドア部分をマウスでクリックすると，建物の内部空間を記述した別のVRMLファイルが呼び出されるようにしている。図3に，建物内部のバーチャルリアリティ空間が呼び出された時の初期画面を示す。これは，廊下のある位置に立って，遠方の廊下の端の方を見ている状態に対応している。廊下の左右には，研究室等が並んでおり，それを示すドアと表札が見えている。ここで，仮想3次元空間の任意の位置に視点を移動して，そこから任意の方向を見ることができるということは，この仮想建物内部を仮想的に歩きまわることが可能であるということを意味している。この操作は，VRMLブラウザでは，空間表示画像部分のマウスによるクリックおよびドラッグ操作で簡単に行われる。図4，5に，廊下を歩いて行くような感じで空間内を移動して行った場合の様子を，スナップショットとして示している。このように見る側の者が，自主的に視点を移動して行くことで，建物内部や部屋配置の様子を現実感を持って知ることができる。図5は，エレベーター施設および階段や別の棟へ通じる廊下等が見えている様子を示している。このようにしながら，各研究室の表札を調べて，ある研究室（ここでは川口研究室とする）にたどり着いて（図6），その研究室を訪問したい時は，ドア部分をマウスでクリックする。すると，クリックした研究室の内部空間を記述したVRMLファイルが呼び出されるようにしてある。図7に研究室内部に入った様子を示す。この部屋の中には1つの机があり，その上にパーソナルコンピューターとモニタが置いた状況になっている。そこで，そのパーソナルコンピューターの前に視点を移動してみる(図8)。ここ

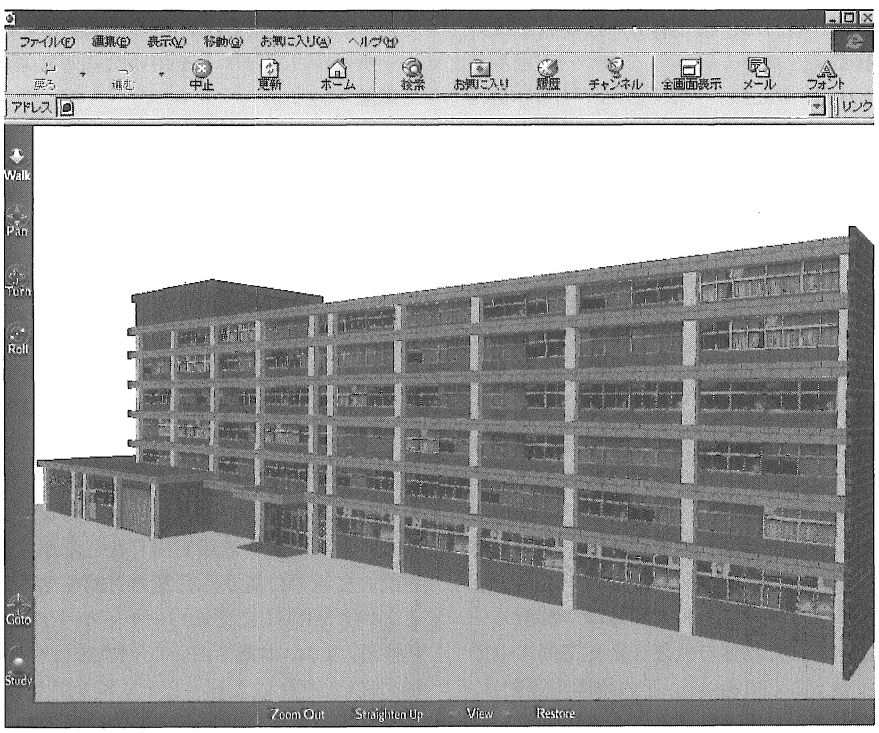

図1　建物の外観

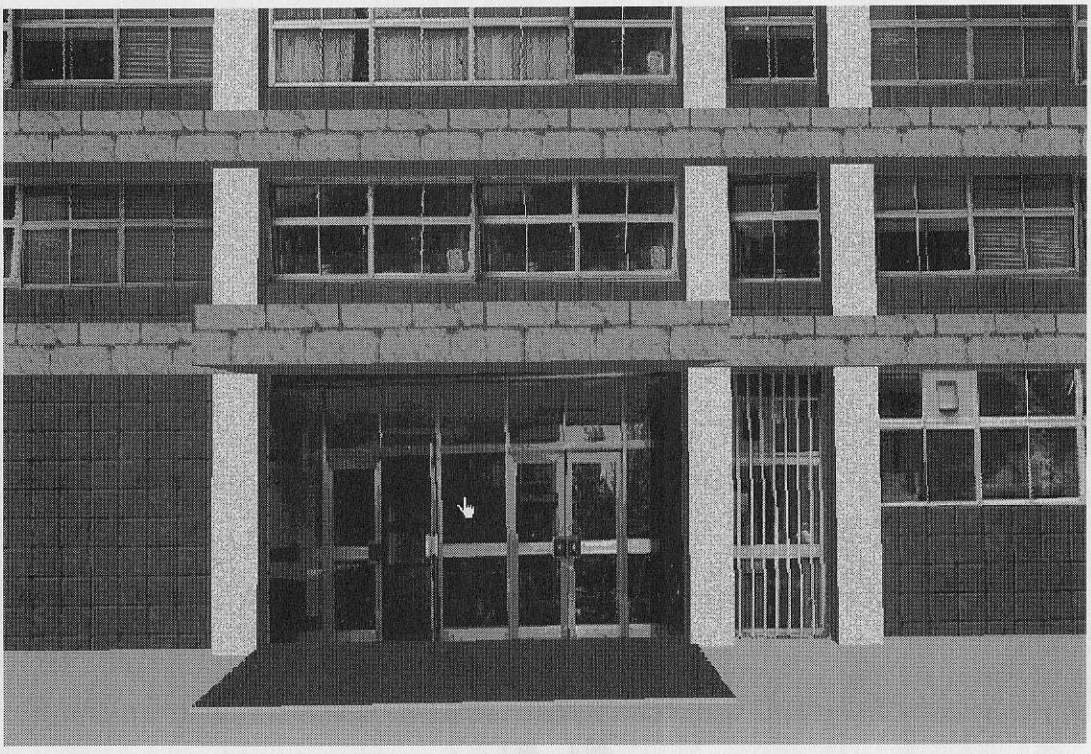

図２　玄関付近

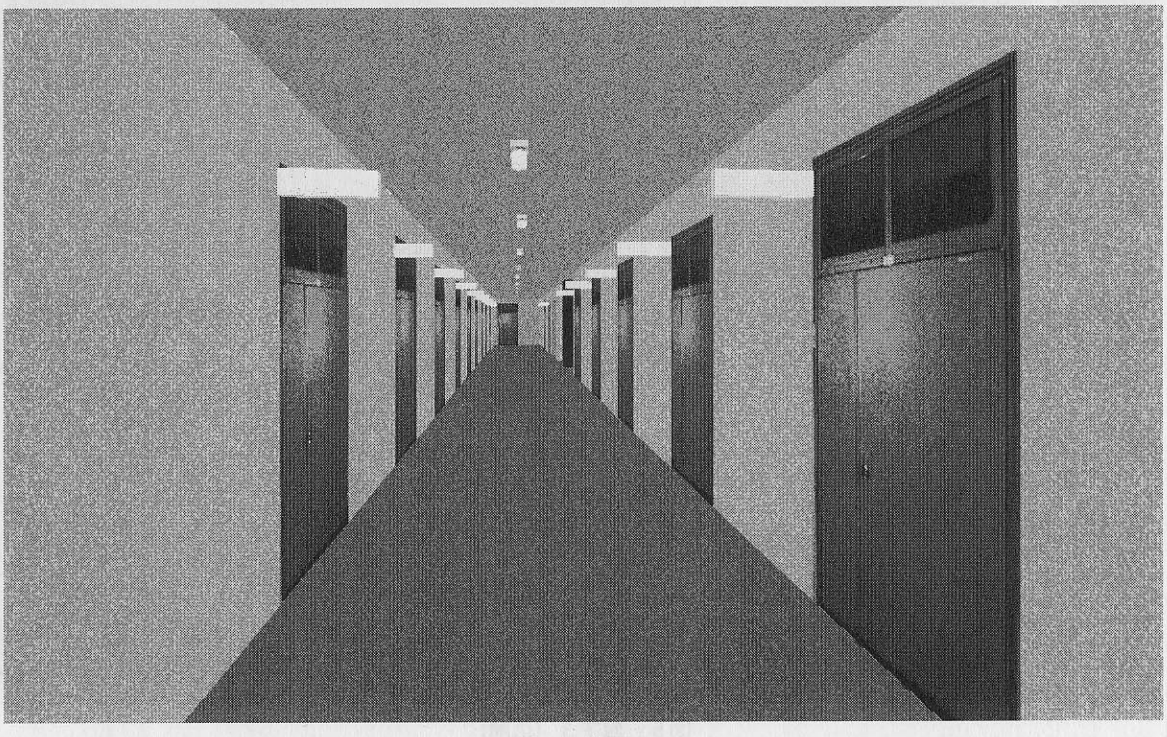

図３　廊下１

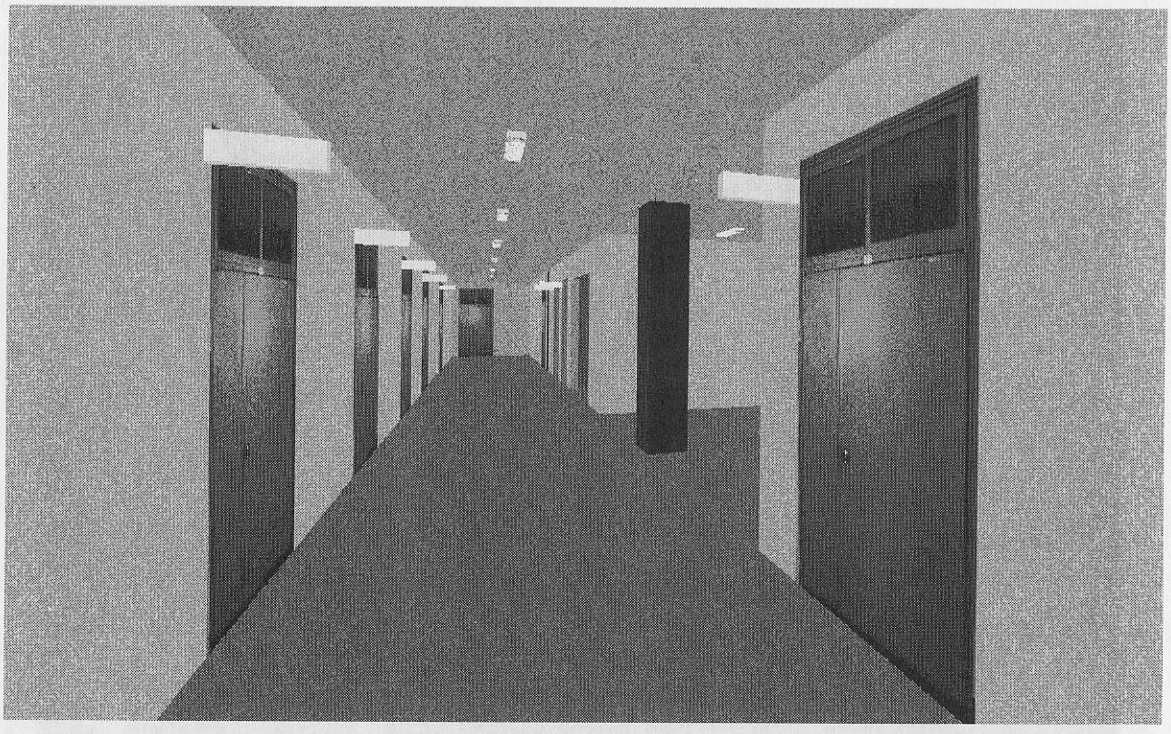

図4　廊下2

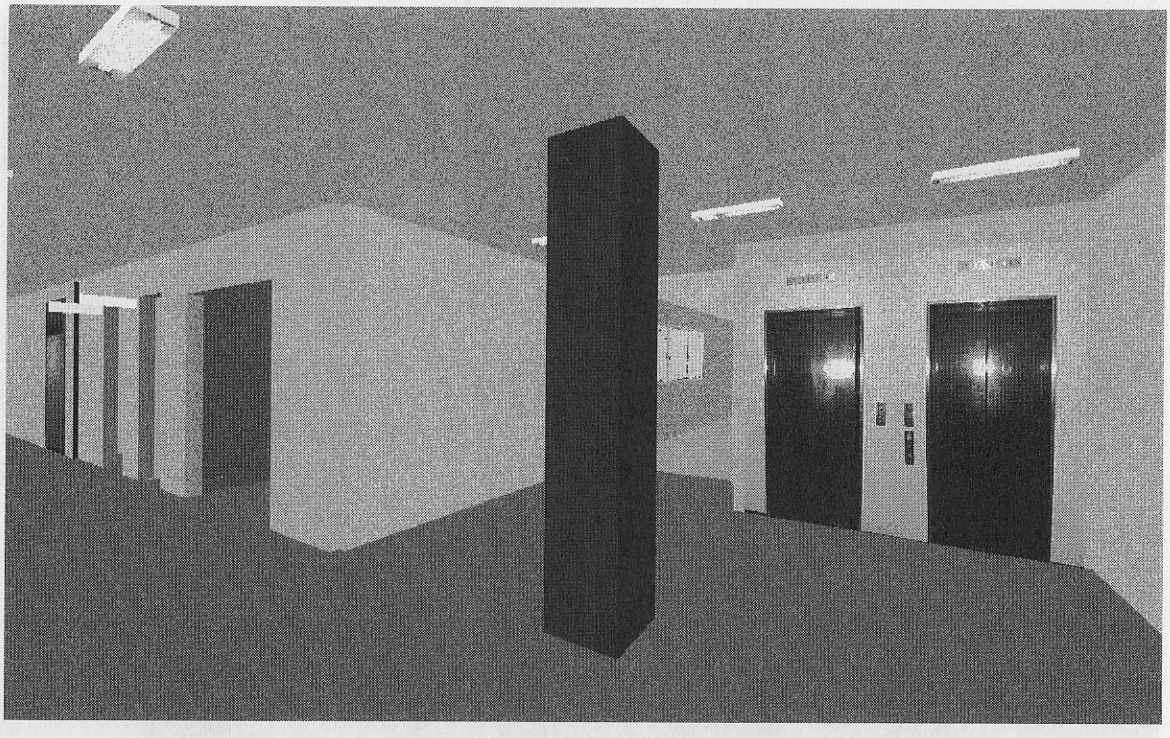

図5　廊下3（広間）

図6　研究室入口ドア前

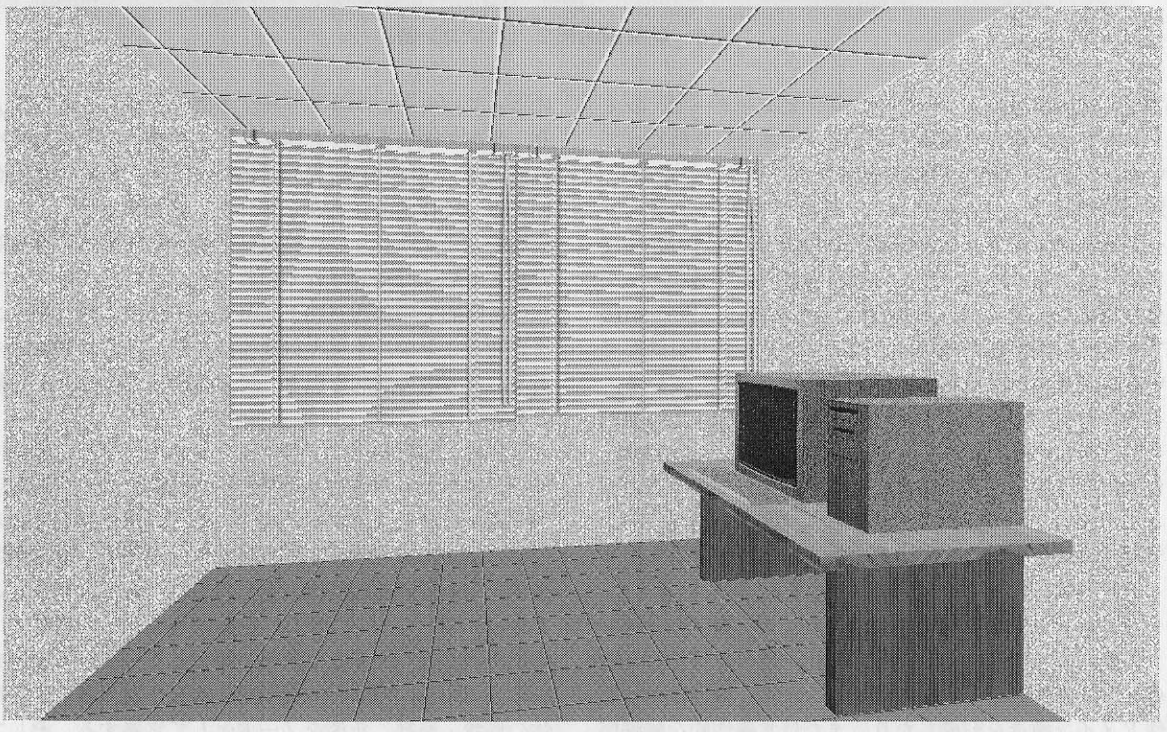

図7　研究室内部1

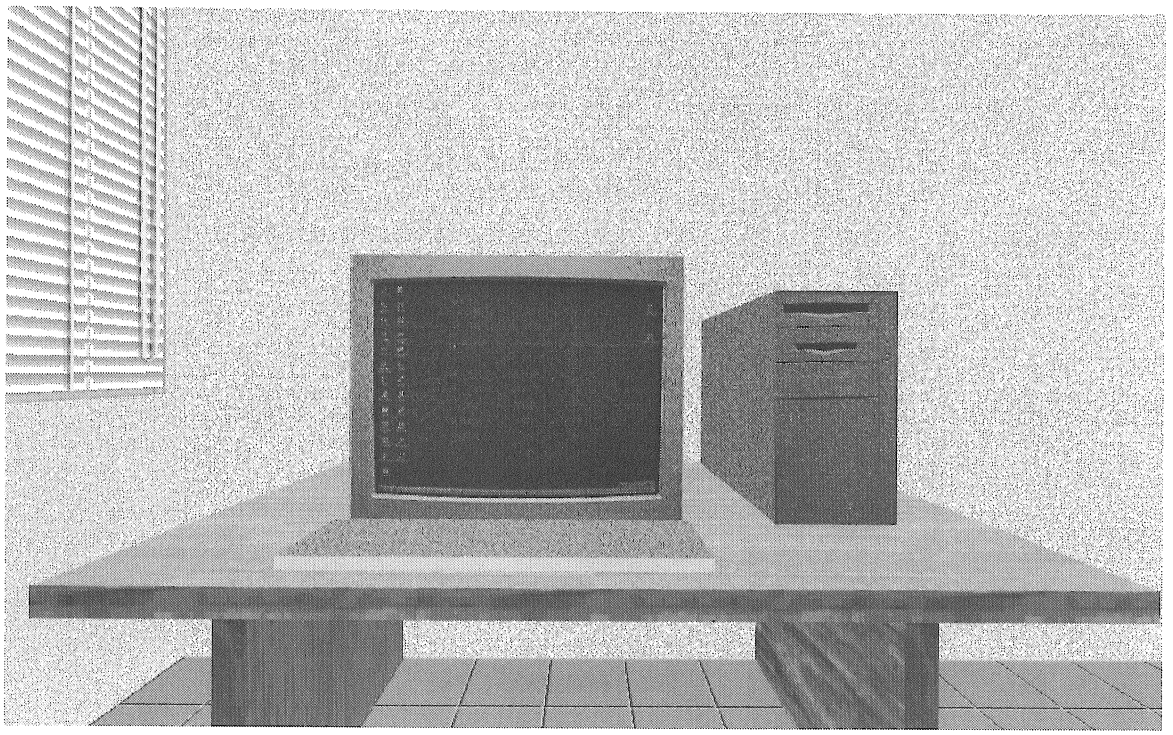

図８　研究室内部２（パソコン前）

では，この仮想的なパーソナルコンピューターで，シミュレーション教材を体験するという意味合いを持たせている。つまり，モニタの画面をマウスでクリックすると，別のVRMLファイルにリンクするのではなく，シミュレーションプログラムを含むHTMLファイルを呼び出すようにリンクしている。ここでは，物質の磁化過程のモンテカルロシミュレーション[3]を行うためのJavaScript[4]のプログラムが記述されたＨＴＭＬファイルにリンクしている。図９には，それを呼び出した画面を示している。このシミュレーションでは，磁性体内部の格子上に並べられたスピン磁気モーメントに対応する小さな正方形が，色付けして配列されている。そして，入力フォームにシミュレーションに必要なパラメータ（温度，磁場，モンテカルロステップ数）の値を入れて，実行ボタンをクリックすれば，シミュレーションが始まる。図10のように，シミュレーション実行過程におけるスピン磁気モーメントの配向が色で区別され，配向変化が色分けパターン変化として理解できる。このパターンから，磁性体が強磁性ドメイン構造を形成したり，または常磁性的な無秩序構造を示したりすることが学習できる。このように，VRMLによってバーチャルリアリティ空間を作成することにより，学部の建物全体から始まって，内部の研究室配置や各研究室レベルでの研究教材紹介までが，階層的に整合性をもって行えることが示された。研究教材は，このようなシミュレーション教材以外でも，WWWブラウザ等で扱える形式で用意されていれば良い。バーチャルリアリティ空間で学部建造物を構築し，その中に教材をリンクさせることで，学部や研究室紹介を，非常に新鮮味のある，見る者の興味を引きつけるものにすることが可能となる。

図11に，今回作成したVRMLファイルの中から建物内部を記述したものを取り上げ，その内容を示す。VRMLファイルはテキストファイルであり，非常に長大であるため，主要部分を抜粋して説明する。図の中で横方向に破線を引いている箇所は，そこに省略された行があることを示している。ここでは，前後の記述から判断して，類似の繰り返しや単なる数値の羅列となっている箇所を省略している。VRMLでは仮想３次元空間に表示する物体の形状，座標，色等をノードとして定義する。そしてノードの性質の詳細をフィールドで定義する。図11に書

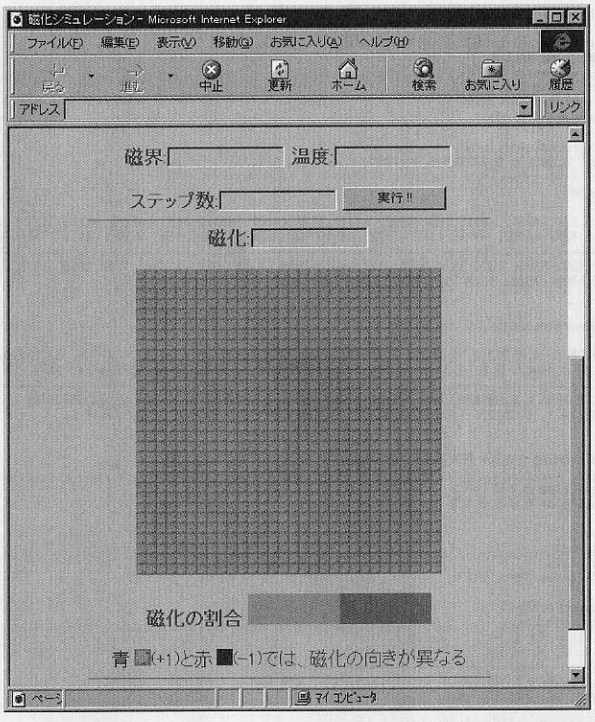

図9　シミュレーション教材（起動時）

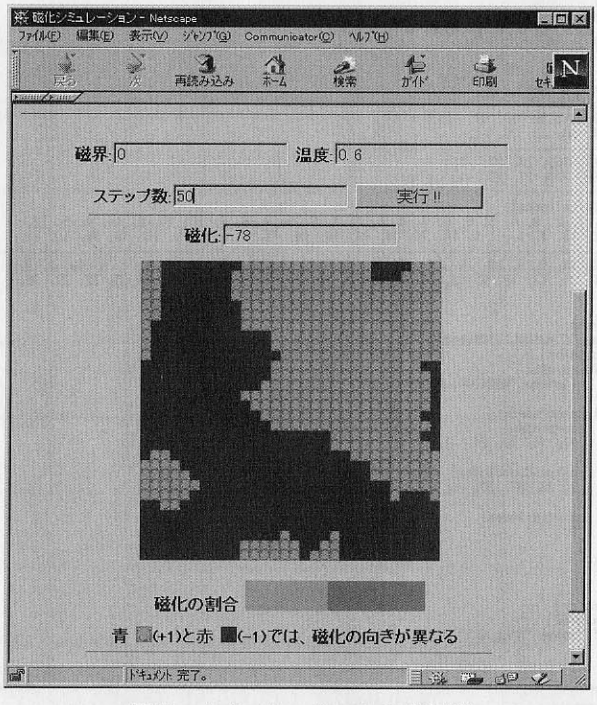

図10　シミュレーションの実行例

```
#VRML V1.0 ascii
Separator{
  ShapeHints {shapeType SOLID vertexOrdering COUNTERCLOCKWISE}
  SpotLight  {on FALSE}
  PerspectiveCamera{
    orientation 0 1 0 1.5708
    position 7544 548 -2449
    heightAngle 0.753901
  }
  DEF BackgroundColor Info{
    string" 0.988235 0.972549 0.941176"
  }
  Coordinate3{
    point[
     -4115 130 -4281, -4115 130 -4832, -4115 1181 -4832, -4115 1181 -4281, -4115 1181 -3772,
     -4115 1277 -4832, -4115 1277 -3772, -4115 130 -3692, -4115 130 -3772, -4115 1277 -3692,
     1312 130 -8575, 1312 597 -8575, 1312 597 -7615, 1312 1099 -11289, 1312 1099 -8575,
     1312 1099 -7612, 1312 597 -7612, 6860 143 -3680,
    ]
  }
  TextureCoordinate2{
    point[
     2.30078 0, 4.45312 0, 4.45312 4.10547, 2.30078 4.10547, 0.3125 4.10547, 4.45312 4.48047,
     0.3125 4.48047, 0 0, 0.3125 0, 0 4.48047, 4.70312 0, 4.77734 0, 4.77734 4.48047,
     0 0.909185, 0 0.966976, 0.00851064 0.966976, 0 0.998968, 0.001221 0, 0.998779 0,
     0.998779 1, 0.001221 1, 0.00952381 0, 0.00952381 1, 0.984416 0, 0.984416 1,
    ]
  }
  MaterialBinding {value PER_FACE_INDEXED}
  Material{
    ambientColor 0.2 0.2 0.2
    diffuseColor 0.8 0.8 0.8
  }
  Separator{
    Separator{
      Texture2{
        filename "WALL.GIF"
      }
      IndexedFaceSet{
        coordIndex[
         0, 1, 2, 3, -1,
         4, 2, 5, 6, -1,
         24, 14, 23, 25, -1,
         26, 27, 28, 29, -1,
        ]
        textureCoordIndex[
         0, 1, 2, 3, -1,
         4, 2, 5, 6, -1,
         7, 14, 23, 24, -1,
         7, 25, 26, 27, -1,
        ]
        materialIndex[
         0, 0, 0, 0, 0, 0, 0, 0, 0,
        ]
      }
    }
    IndexedFaceSet{
      coordIndex[
       30, 31, 32, 33, -1,
       34, 30, 33, 35, -1,
       770, 769, 28, 27, -1,
       768, 770, 27, 26, -1,
      ]
      materialIndex[
       1, 2, 2, 3, 4, 5, 6, 6, 7, 8, 8, 8, 8, 9, 9, 9, 9, 6, 10, 10, 6, 11, 6, 6, 12,
       12, 13, 1, 1, 1, 1, 14, 14, 14, 14, 14, 14, 11, 11, 11, 15, 16, 16, 16, 16, 11,
       30, 24, 5, 5, 23, 19, 23, 23, 11, 22, 15, 15, 15, 15, 24, 4, 4, 4, 3, 19, 2, 2,
       31, 31, 12, 12, 6, 13, 28, 7, 8, 20, 23, 31, 31, 31, 31, 31, 12, 6, 32, 27, 26,
      ]
    }
    WWWAnchor{
      name "C:¥SIMULATION.html"
      Separator{
        Texture2{
          filename "MONITOR.JPG"
        }
        IndexedFaceSet{
          coordIndex[
           17, 16, 20, 21, -1,
          ]
          textureCoordIndex[
           7, 28, 29, 30, -1,
          ]
          materialIndex[
           0,
          ]
        }
      }
    }
    Separator{
      Texture2{
        filename "WALL.GIF"
```

図11－1　建物内部を記述したVRMLファイル（その1）

```
    }
    IndexedFaceSet{
      coordIndex[
       771, 772, 773, 774, -1,
       18, 775, 776, 22, -1,
      ]
      textureCoordIndex[
       7, 31, 32, 33, -1,
       7, 31, 34, 24, -1,
      ]
      materialIndex[
       0, 0,
      ]
    }
  }
-----------------------------------------------------------
  Separator{
    Texture2{
      filename "COMPUTER.JPG"
    }
    IndexedFaceSet{
      coordIndex[
       772, 785, 788, 789, -1,
      ]
      textureCoordIndex[
       7, 28, 29, 30, -1,
      ]
      materialIndex[
       0,
      ]
    }
  }
-----------------------------------------------------------
  Separator{
    Texture2{
      filename "DESK.GIF"
    }
    IndexedFaceSet{
      coordIndex[
       793, 794, 751, 762, -1,
       764, 760, 795, 765, -1,
      ]
      textureCoordIndex[
       7, 63, 64, 65, -1,
       7, 66, 67, 68, -1,
      ]
      materialIndex[
       0, 0,
      ]
    }
  }
-----------------------------------------------------------
  Separator{
    Texture2{
      filename "WALL2.GIF"
    }
    IndexedFaceSet{
      coordIndex[
       700, 811, 740, 696, -1,
       812, 746, 811, 813, -1,
      ]
      textureCoordIndex[
       97, 98, 99, 100, -1,
       7, 101, 100, 102, -1,
      ]
      materialIndex[
       0, 0,
      ]
    }
  }
-----------------------------------------------------------
  Separator{
    Texture2{
      filename "LIGHT.JPG"
    }
    IndexedFaceSet{
      coordIndex[
       357, 356, 360, 362, -1,
       365, 364, 368, 370, -1,
-----------------------------------------------------------
       657, 656, 660, 662, -1,
       665, 664, 668, 670, -1,
      ]
      textureCoordIndex[
       7, 209, 210, 211, -1,
       7, 212, 213, 214, -1,
-----------------------------------------------------------
       7, 215, 216, 217, -1,
       7, 218, 219, 220, -1,
      ]
      materialIndex[
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      ]
    }
  }
-----------------------------------------------------------
  Separator{
    Texture2{
      filename "DOOR.JPG"
    }
    IndexedFaceSet{
      coordIndex[
       8, 0, 3, 4, -1,
      ]
      textureCoordIndex[
       7, 244, 245, 30, -1,
      ]
      materialIndex[
       0,
      ]
    }
  }
-----------------------------------------------------------
  Separator{
    Texture2{
      filename "DOOR.JPG"
    }
    IndexedFaceSet{
      coordIndex[
       1169, 1166, 1168, 1170, -1,
       1178, 1172, 1175, 1176, -1,
       1170, 1168, 1172, 1178, -1,
      ]
      textureCoordIndex[
       7, 755, 756, 757, -1,
       758, 759, 760, 30, -1,
       757, 756, 759, 758, -1,
      ]
      materialIndex[
       0, 0, 0,
      ]
    }
  }
-----------------------------------------------------------
  Separator{
    Texture2{
      filename "ELEVATOR.JPG"
    }
    IndexedFaceSet{
      coordIndex[
       1180, 1145, 1148, 1149, -1,
       1181, 1150, 1145, 1180, -1,
      ]
      textureCoordIndex[
       769, 770, 29, 30, -1,
       7, 28, 770, 769, -1,
      ]
      materialIndex[
       0, 0,
      ]
    }
  }
-----------------------------------------------------------
  Separator{
    Texture2{
      filename "WINDOW.JPG"
    }
    IndexedFaceSet{
      coordIndex[
       1198, 1193, 1196, 1197, -1,
      ]
      textureCoordIndex[
       7, 28, 800, 801, -1,
      ]
      materialIndex[
       0,
      ]
    }
  }
-----------------------------------------------------------
  Separator{
    Texture2{
      filename "LAB.JPG"
    }
    IndexedFaceSet{
      coordIndex[
       291, 286, 288, 289, -1,
      ]
      textureCoordIndex[
       807, 808, 809, 810, -1,
      ]
      materialIndex[
       0,
      ]
    }
  }
}
```

図11－2　建物内部を記述したVRMLファイル（その2）

かれている主なノードについて以下に説明する。Separator { } は一つのノードグループを形成し，その中での座標系等は他のグループとは独立で影響されない。そして，そのグループ内のノードTexture2{filename "..."}では，バーチャルリアリティ空間に表示するための画像ファイルを指定してる。例えば，WALL.GIF(46行目)，MONITOR.JPG (102行目)，COMPUTER.JPG (139行目)，DOOR.JPG (225行目) およびELEVATOR.JPG (265行目) は，それぞれ，上述のバーチャルリアリティ空間における壁の模様の一つ，研究室内部にあったパーソナルコンピュータのモニタ正面，コンピュータの正面，廊下から見た研究室のドア，およびエレベーターの扉についての画像に対応している。Material { } は物体の表面の色や透明度に関する材質を定義するノードである。Coordinate3 { } は各ノードで使われる3次元座標の集合を定義するノードである。つまり，建物内部構造を記述するのに必要なすべての座標を指定するために，かなり膨大な数値列がここで指定されている。TextureCoordinate2 { } は画像を物体に貼り付けて表示させるためのノードである。IndexedFaceSet{ }は空間内の面を，coorIndex[ ]等のフィールドを用いて定義するノードである。WWWAnchor { } というノード内では，name "C:¥SIMULATION.html"と記述することで，図9で示したシミュレーションプログラムを含むHTMLファイルを指定しており，これとモニタ画像ファイル(MONITOR.JPG)とのリンクがなされている。

3．バーチャルリアリティを利用した科学技術教育用シミュレーション教材の試作

本章ではバーチャルリアリティを用いた科学技術教育用シミュレーション教材[1]についての説明と議論を行う。シミュレーション対象として，3次元空間に存在する荷電粒子集団を考える。これは実際には，半導体デバイス内におけるキャリア集団[5]や，放電現象における電離気体等が該当する。そして，個々の粒子の運動を記述する運動方程式の時間発展を計算することでシミュレーションを行い，同時に各粒子の仮想3次元空間内での座標値を変更してゆき，シミュレーション結果のリアルタイム3次元可視化を実現する。ここで，バーチャルリアリティ空間の構築には，先と同様にVRMLを用い，粒子の運動の時間発展を計算するプログラムはJava言語で記述した。このような粒子集団の運動の時間発展を調べる方法は，分子動力学法[6,7]と呼ばれ，モンテカルロ法[6,7,8]等と共に代表的な科学技術用シミュレーション技法として知られている。また，3次元可視化技術は，現在，非常に精力的に研究が進められており，単なる数値の羅列であるシミュレーション結果に対して物理的洞察を行う際に，必要不可欠である。そこで，本教材を用いてシミュレーションの実行から結果の確認までを，リアルタイムで3次元可視化された状態で学習することは，上述の半導体デバイス内におけるキャリア等に関する現象・理論を理解する上で有益であるのみならず，科学技術計算についての認識を深めることにもつながっている。

考慮する対象は荷電粒子であるので，粒子間相互作用としてクーロン型ポテンシャル$U(r)=q(a/r)^2$を採用した。ここで，rは粒子間距離，aは粒子間距離の規格化因子，qは粒子の電荷の2乗に対応するパラメータである。ここでq，aは，それぞれ，エネルギー，距離の次元を持っている。クーロン型のような長距離型ポテンシャルの場合，本教材の扱うような非常に小規模な系では，1つの粒子は残りのすべての粒子と強く相互作用していることになる。そのため，シミュレーション結果に対する有限サイズ効果は，定量的な議論の際には無視できないが，本教材では定性的な系の振る舞いを議論するので重要な問題とはならない。以下では物理量を無次元化して表すが，温度をT^*で表わす。これはk_BT/qを意味し，k_BとTは，それぞれボルツマン定数と実際の温度である。そして，系の粒子数を固定した条件下で，aの大きさにより平均粒子間距離を変えて，粒子数密度$(N/(V/a^3))$を調整する。NとVは，それぞれ全粒子数と系の体積である。ある温度における粒子速度分布はマクスウェルーボルツマン分布に従うが，その発生にはBox-Muller法[8,9]により生成したガウス分布の乱数を用いている。分子動力学シミュレーションでは，各粒子のニュートン運動方程式の時間発展を時間差分によって，数値的に求めてゆく。そして，ある一定時間おきに，粒子の速度をある一定温度における分布に従うようにスケーリングを行うことで，温度がほぼ一定に保たれるように調整する。[6]また，このスケーリング操作によって，任意の方向へ外力が印加された状況下で，粒子集団の重心は，ほぼ一定の平均速度でドリフトして行く。従って，このシミュレーションでは，粒子集団における外力の影響，濃度拡散，そして熱による擾乱の効果を取り入れていることになる。上記の分子動力学シミュレーションおよびVRMLによる可視化の手順は，以下のようになる。

1．VRMLファイル内で，仮想3次元空間に粒子をY－Z平面に沿って，小さなX座標値のところに並べた状態を初期配置とする。

2．VRMLでの粒子座標値と外力（F）をもとに，Java

のプログラムで，ある時刻における各粒子に働く力の総和を計算する。

3．Javaのプログラムで，ニュートン運動方程式を時間差分して，次の時刻における粒子座標値および速度を全粒子について計算する。これがシミュレーションの1ステップとなる。

4．ある一定時間（ここでは10ステップ）経過する度に，所定の温度T*に対応した速度分布となるように全粒子の速度を調整する。同時に，VRMLで定めた粒子座標値も更新し，ブラウザ上での全粒子の移動を行う。

5．系が平衡状態に達する程度まで，上記の2．から4．の過程を繰り返す。

初期粒子配置を粒子がボックスのある側面付近に並べられた状態にしておく。これは，ある側面から高密度で粒子集団が注入された状況を想定している。半導体デバイスでは，このような状況はキャリアの注入として実現されている。この状態から粒子が，ドリフト・拡散して行く様子をシミュレーションすることにする。なお，ここではVRMLブラウザはSONY製のCommunity Place Browser[10]を用いる。これは，このブラウザはVRMLのJavaプログラムによる制御に正式対応していることから，信頼性が高いと考えられるためである。

図12から15に，初期状態から時間発展と共に粒子が移動して行く様子を，ある3つの時間で観察した連続的スナップショットとして示している。この中で粒子の移動を把握し易くするために，粒子集団の中から2個をマーカー粒子として選び，他の粒子と色を変えて区別している。パラメータ値は，N＝32，V＝2^3，T*＝0.1，q＝1，a＝0.171，F＝0.3としている。ここでは，Fは外

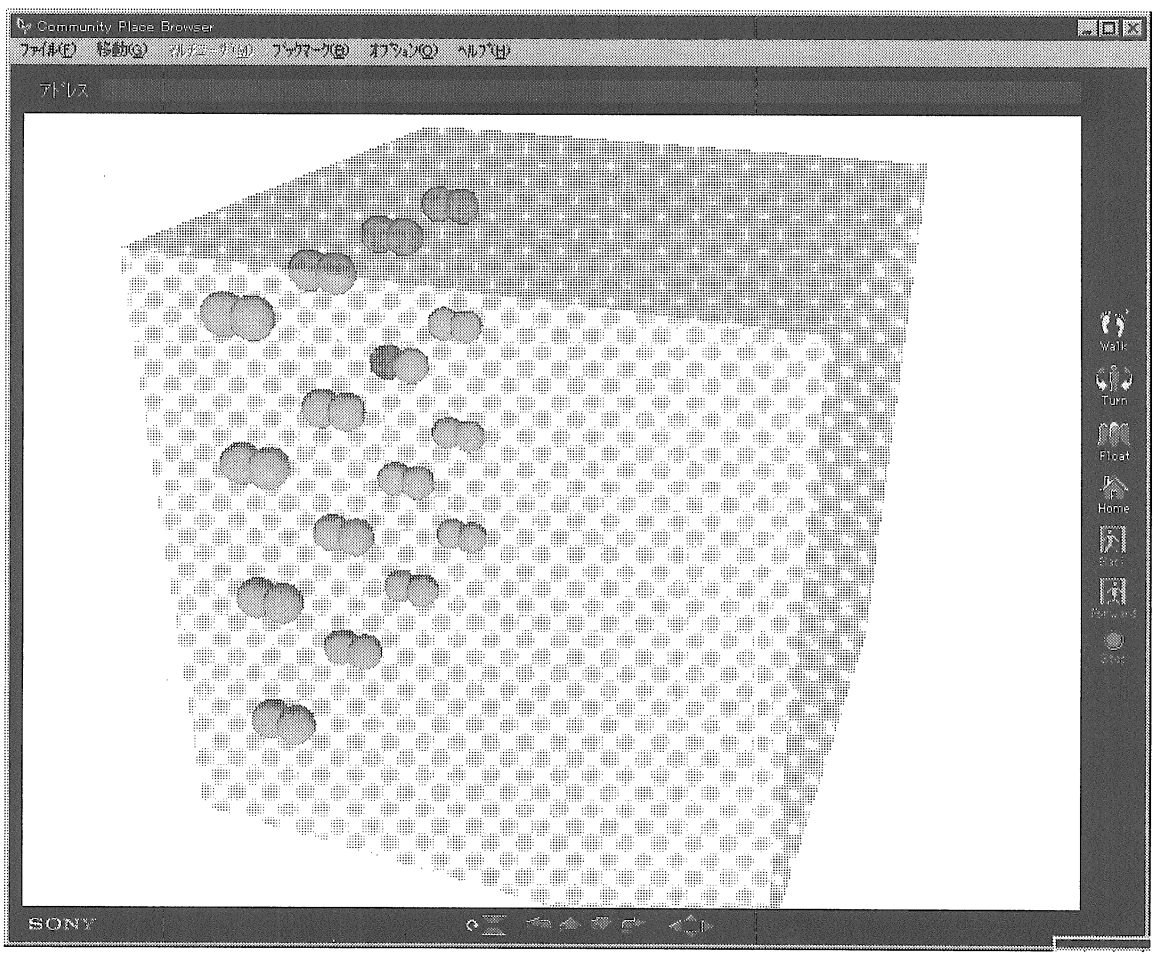

図12　3次元可視化されたシミュレーション教材の全体（実行前の初期配置）

力の絶対値表わしており，初期状態（図12）の局在した粒子集団を強制拡散させる方向（＋X方向）に印加している。本論文では，これらのパラメータ値の変化による定量的な議論は行わないで，粒子の運動の定性的な振る舞いにのみ着目して議論する。はじめ全粒子は，ドリフトして行きながらも，図13，14のように，初期状態を反映して空間的に局在して不均一に分布している。しかし，時間と共に粒子分布は広がり，ドリフトと同時に拡散も生じる。ある程度時間が経って，定常状態に達すると，粒子は均一に分布し，流体的な運動になる(図15)。このような振る舞いは，相互作用の強さと粒子密度に強く依存する。ここの図に示した状態は，相互作用が比較的弱く，低粒子密度の場合であるので，粒子の全体的運動は，空間的な規則構造は持たない構造のドリフトという，流体的運動になっている。逆に，相互作用を強く，高粒子密度にすると，粒子集団はある程度の空間的規則性のある構造を持ちながらのドリフト運動を示すようになる。これは流体運動と言うよりも，非常に強い相互作用するクラスタ状の粒子集団のドリフトというような非現実的な特殊な状態である。上述のように，このようなシミュレーションに対応する実際の現象としては，トランジスタ等の半導体デバイス内におけるキャリアのドリフト・拡散等が対応する。ただし，粒子数はシミュレーション中，常に一定であり，電子－正孔相互作用により生じるような粒子の生成消滅の効果は，ここでは考慮できない。また，他の対応するものとして，粒子間相互作用をレナードージョーンズポテンシャルのようなタイプに変更すれば，熱流体系[11]を想定することができ，分子レベルでの運動のシミュレーションが可能である。

ここで，VRMLによるバーチャルリアリティ空間で，

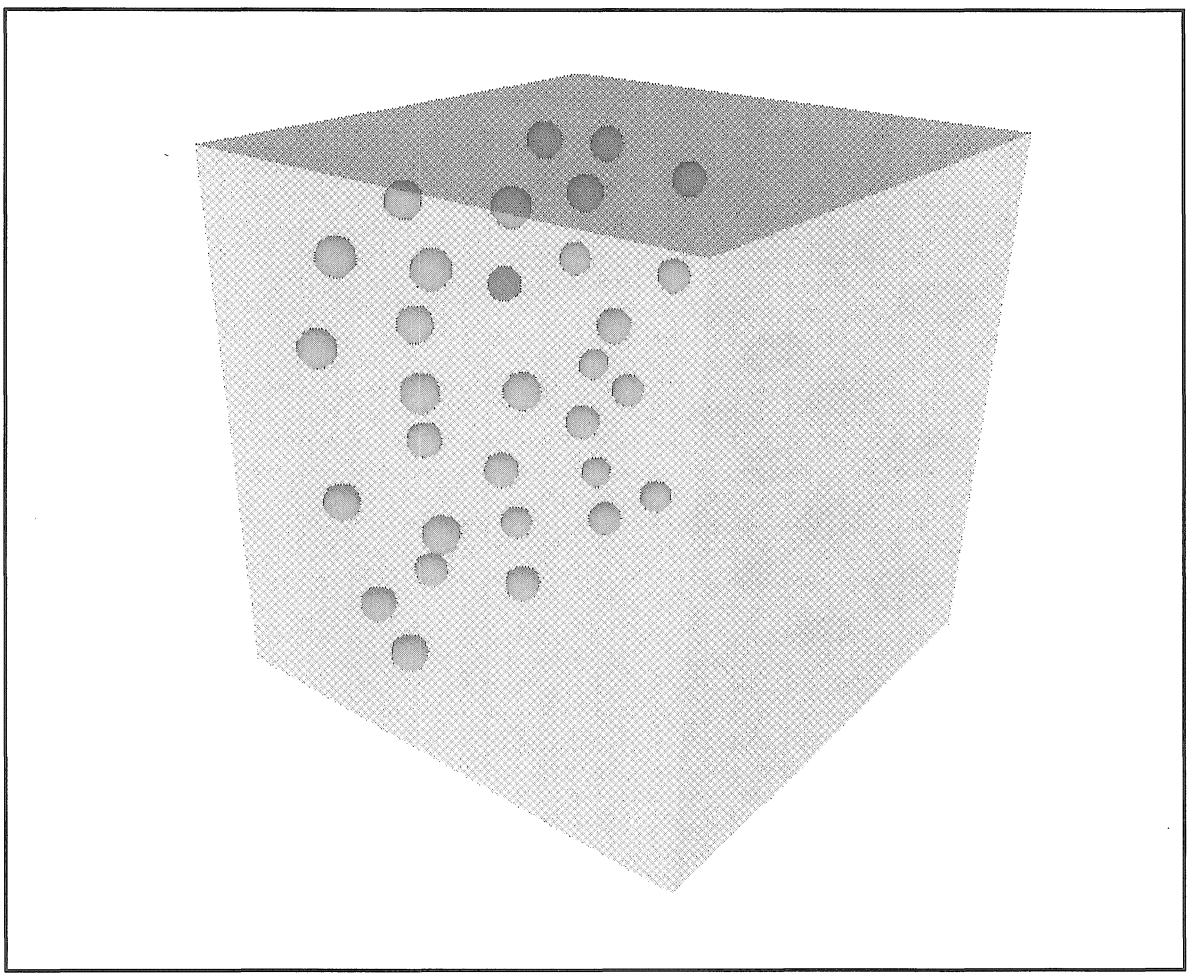

図13　シミュレーション実行結果（初期段階その１）

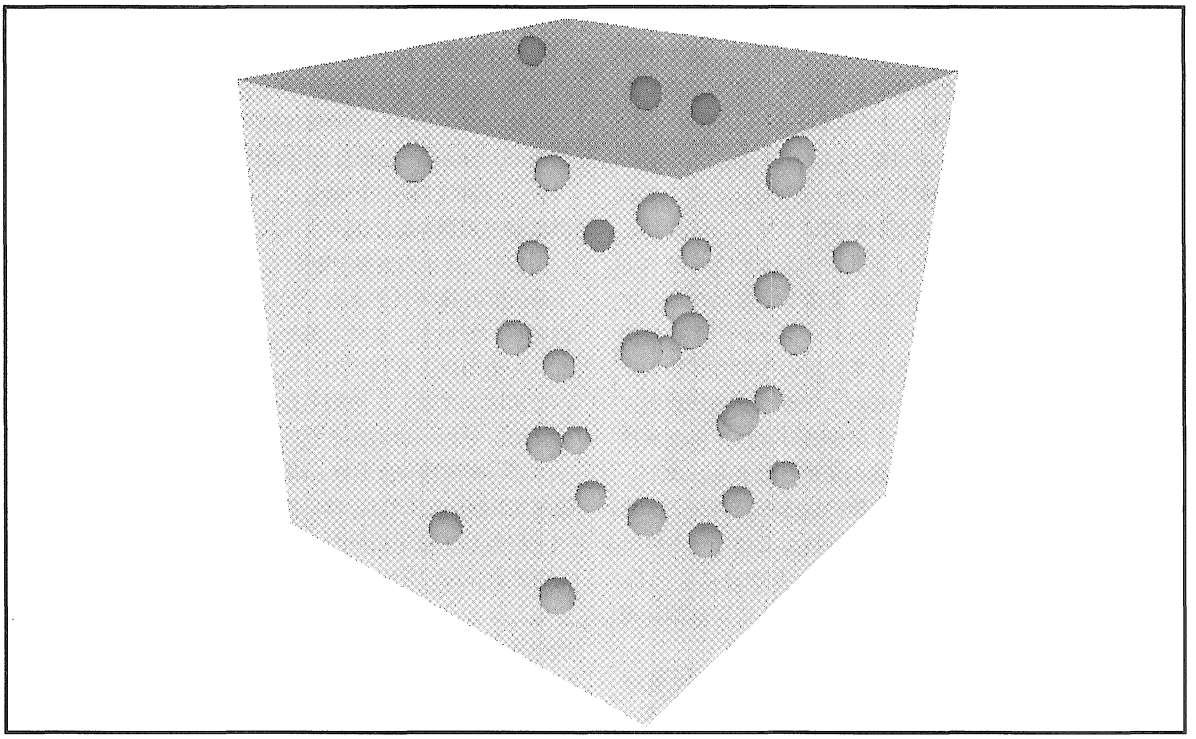

図14　シミュレーション実行結果（初期段階その２）

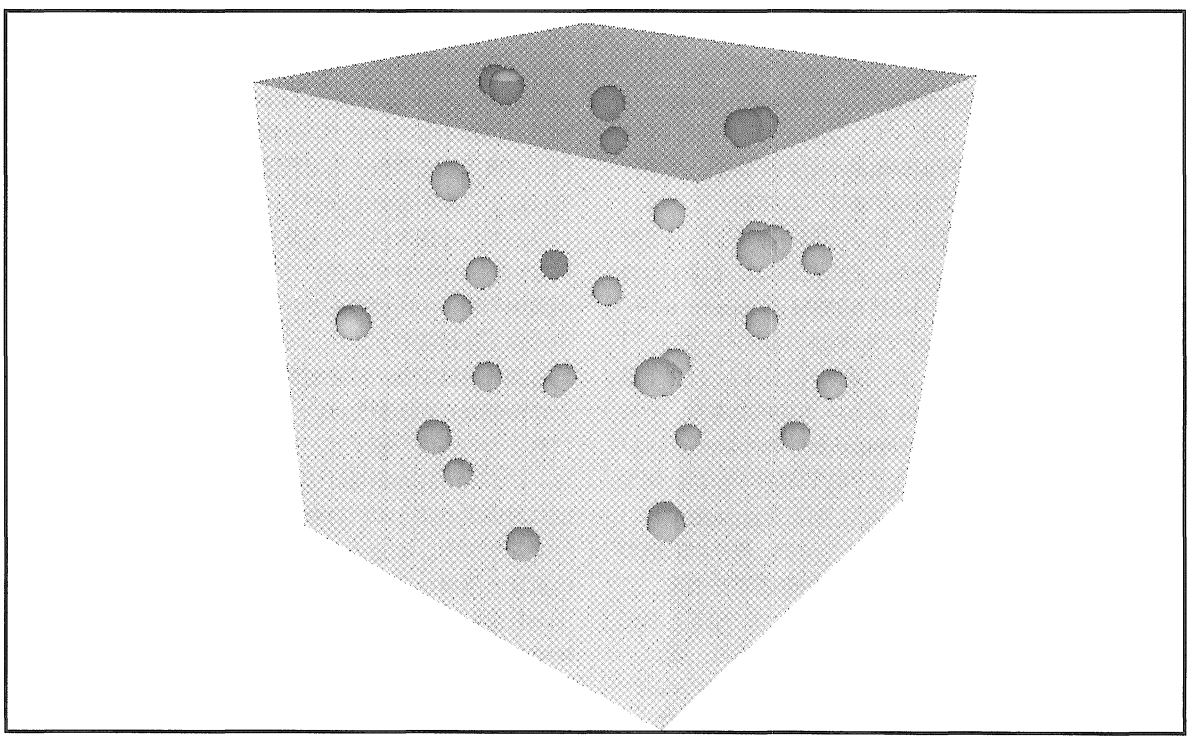

図15　シミュレーション実行結果（定常状態）

このようなシミュレーションを実行することの最大の利点は，先の学部・研究室紹介の場合と同様に，実行中の途中結果を任意の位置と角度で観察することが可能であるということにある。従来のシミュレーション教材では，3次元可視化をすることが可能ではあっても，ある定まった角度や位置でしか観測できないものがほとんどであった。シミュレーション結果の観測における利便さは，本教材のように粒子集団による3次元的な構造を調べるという問題意識のある場合には，重要な要素となる。本研究で開発した教材では，シミュレーションの実行に関するパラメータの入力方法として，Javaのプログラムに直接書き込むようになっているが，ブラウザ上に数値入力フォーム等のインターフェイスを設ければ，より簡単に温度や密度等のパラメータ値の変更ができる。また，粒子数を増やして，もう少し規模の大きな系のシミュレーションを行えるように，プログラムまたはハードウェア上の工夫をする余地がある。このような改良によって，各パラメータの値に応じた粒子の振る舞いの定量的な議論が可能になるであろう。

図16に，バーチャルリアリティ空間に粒子集団を記述したVRMLファイル（particle.wrl）の内容[1]を示す。この2から4行目までのBackground {・・} では，背景色をRGBの3原色で指定している。5から8行目までのViewpoint {・・} は，バーチャルリアリティ空間で物体を眺める際の視点の座標，角度を指定している。9から23行目までは，Tramsform {・・Shape {geometry Box { }・・}・・} によって，シミュレーションで考慮している粒子の運動可能な領域とその形状を，半透明色の立方体のボックスで表示するように指定している。次に，ボックスの中に表示させる32個の粒子について形状等を指定する必要があるが，そのために，まず1つの粒子を，24から39行目で，DEF ATOM1 Transform{・・DEF Atom Shape {geometry Sphere {radius 0.07}}・・} として定義している。ここで，DEF ATOM1とは，1番目の粒子（ATOM1）を定義するという意味であり，その形状（geometry）を，半径（radius）が0.07の球（Sphere）として指定している。そして，このATOM1の形状に関するデータを受け継いで，DEF ATOM2 Transform { } からDEF ATOM32 Transform { } で，残りの31個の粒子の形状と新たにそれぞれの初期配置座標を指定している。途中のATOM10とATOM19をマーカー粒子として定めて，色を他の粒子のものとは変更している。次に，246から283行にかけてのDEF MoveIt Script {・・url "particle.class"・・field SFNode node1 USE ATOM1・・} で，32個の粒子の運動を制御するためのJava言語による計算プログラムを中間コードに変換したクラスファイル（ファイル名：particle.class）を指定している。また，ここで，各粒子の座標値データをノード（SFNode）として計算プログラムに受け渡すための指定も行っている。最後に，284から288行目で，DEF MyTimer TimeSensor {・・} ROUTE・・の部分で，ある一定時間周期で粒子座標値を更新を繰り返すように指示している。

図17に，粒子の運動を制御しているJava言語による分子動力学計算プログラム（ファイル名：particle.java）の内容1）を示す。まず，最初の部分で，変数の定義とそれへの値の代入を行っている。以下に，変数の意味について説明しておく。全粒子数はN（=32），系の形状は立方体で，その一辺の長さをlleng（=2.0），シミュレーションでの時間積分における時間の刻み幅をｄｔ（=0.03），温度T*をtref（=0.1），粒子間距離の規格化因子ａをsigm（=0.171），粒子に印加する外力FのX，Y，Z成分をそれぞれ，efx（=0.3），efy（=0），efz（=0）としている。シミュレーションのパラメータ値に関する条件を変更するには，これらの変数への代入値を変えて，再度，クラスファイルを生成し直すことになる。また，粒子間相互作用により作用する力のX，Y，Z成分に対応する変数を，それぞれｆｘ，ｆｙ，ｆｚとしている。粒子のX，Y，Z座標および速度成分に対する変数は，それぞれxyzとuvwと定義している。42から52行目のpublic void processEvent（Event e）{・・} の{ } 中に，上記のVRMLファイル（particle.wrl）から一定時間周期で呼び出して実行するプログラムを記述する。ここでは，{ } 内には実行する関数のみを指示しており，その関数の内容は以下の別の部分で記述している。以下に，ここで用いられている関数について順に説明する。inptvrml( )（211から298行目で記述）では，バーチャルリアリティ空間内の粒子の座標値を受け取って，それをプログラム中の粒子座標変数ｘ，ｙ，ｚに代入する。force( )（157から201行目）では，粒子座標から粒子間力を計算し，それに外部からの駆動力efx，efy，efzを合わせて，各粒子に働く力の各成分値ｆｘ，ｆｙ，ｆｚを計算する。vscale( )（73から139行目）では，粒子速度分布が所定の温度に対応するように，各粒子速度をある一定割合でスケーリングする。mov( )（141から155行目）では，粒子の速度と座標から，次の単位時間ステップ経過後に，粒子が移動して存在しているべき座標値を計算する。また同時に，速度に相当する量を求める箇所があるが，これは単位時間ステップ経過後の速度の計算のために必要な途中の計算式に対応している。この計算方法は，運動方程式の時間発展を数値的に求める際の差分化の方法に関

```
#VRML V2.0 utf8
Background {
  skyColor 1 1 1
}
Viewpoint {
  position 3.6 2.85 4.5
  orientation -0.1 0.056 -0.1 1.1
}
Transform {
  translation 1 1 1
  children [
    Shape {
      geometry Box {}
      appearance Appearance {
        material Material {
          diffuseColor 0.501961 0.478431 0.470588
          emissiveColor 0.105882 0.211765 0.12549
          transparency 0.5
        }
      }
    }
  ]
}
DEF ATOM1 Transform {
  translation 0.15 0.25 0.25
  children [
    DEF Atom Shape {
      geometry Sphere {
        radius 0.07
      }
      appearance Appearance {
        material DEF SphereColor Material {
          diffuseColor 0.45098 1 0.168627
          emissiveColor 0.270588 0.270588 0.270588
        }
      }
    }
  ]
}
DEF ATOM2 Transform {
  translation 0.15 0.25 0.75
  children [
    USE Atom
  ]
}
DEF ATOM3 Transform {
  translation 0.15 0.25 1.25
  children [
    USE Atom
  ]
}
DEF ATOM4 Transform {
  translation 0.15 0.25 1.75
  children [
    USE Atom
  ]
}
DEF ATOM5 Transform {
  translation 0.15 0.75 0.25
  children [
    USE Atom
  ]
}
DEF ATOM6 Transform {
  translation 0.15 0.75 0.75
  children [
    USE Atom
  ]
}
DEF ATOM7 Transform {
  translation 0.15 0.75 1.25
  children [
    USE Atom
  ]
}
DEF ATOM8 Transform {
  translation 0.15 0.75 1.75
  children [
    USE Atom
  ]
}
DEF ATOM9 Transform {
  translation 0.15 1.25 0.25
  children [
    USE Atom
  ]
}
DEF ATOM10 Transform {
  translation 0.15 1.25 0.75
  children [
    DEF Atom2 Shape {
      geometry Sphere {
        radius 0.07
      }
      appearance Appearance {
        material DEF SphereColor Material {
          diffuseColor 0.9 0 0.5
          emissiveColor 0.270588 0.270588 0.270588
        }
      }
    }
  ]
}
DEF ATOM11 Transform {
  translation 0.15 1.25 1.25
  children [
    USE Atom
  ]
}
DEF ATOM12 Transform {
  translation 0.15 1.25 1.75
  children [
    USE Atom
  ]
}
DEF ATOM13 Transform {
  translation 0.15 1.75 0.25
  children [
    USE Atom
  ]
}
DEF ATOM14 Transform {
  translation 0.15 1.75 0.75
  children [
    USE Atom
  ]
}
DEF ATOM15 Transform {
  translation 0.15 1.75 1.25
  children [
    USE Atom
  ]
}
DEF ATOM16 Transform {
  translation 0.15 1.75 1.75
  children [
    USE Atom
  ]
}
DEF ATOM17 Transform {
  translation 0.25 0.25 0.25
  children [
    USE Atom
  ]
}
DEF ATOM18 Transform {
  translation 0.25 0.25 0.75
  children [
    USE Atom
  ]
}
DEF ATOM19 Transform {
  translation 0.25 0.25 1.25
  children [
    DEF Atom3 Shape {
      geometry Sphere {
        radius 0.07
      }
      appearance Appearance {
        material DEF SphereColor Material {
          diffuseColor 0.698039 0.388235 0.529412
          emissiveColor 0.270588 0.270588 0.270588
        }
      }
    }
  ]
}
DEF ATOM20 Transform {
  translation 0.25 0.25 1.75
  children [
    USE Atom
  ]
}
DEF ATOM21 Transform {
  translation 0.25 0.75 0.25
  children [
    USE Atom
  ]
}
DEF ATOM22 Transform {
  translation 0.25 0.75 0.75
  children [
    USE Atom
  ]
}
DEF ATOM23 Transform {
  translation 0.25 0.75 1.25
  children [
    USE Atom
  ]
}
DEF ATOM24 Transform {
  translation 0.25 0.75 1.75
  children [
    USE Atom
  ]
}
DEF ATOM25 Transform {
```

図16－1　シミュレーション教材のVRMLファイル（その1）

```
199   translation 0.25 1.25 0.25
200   children [
201     USE Atom
202   ]
203 }
204 DEF ATOM26 Transform {
205   translation 0.25 1.25 0.75
206   children [
207     USE Atom
208   ]
209 }
210 DEF ATOM27 Transform {
211   translation 0.25 1.25 1.25
212   children [
213     USE Atom
214   ]
215 }
216 DEF ATOM28 Transform {
217   translation 0.25 1.25 1.75
218   children [
219     USE Atom
220   ]
221 }
222 DEF ATOM29 Transform {
223   translation 0.25 1.75 0.25
224   children [
225     USE Atom
226   ]
227 }
228 DEF ATOM30 Transform {
229   translation 0.25 1.75 0.75
230   children [
231     USE Atom
232   ]
233 }
234 DEF ATOM31 Transform {
235   translation 0.25 1.75 1.25
236   children [
237     USE Atom
238   ]
239 }
240 DEF ATOM32 Transform {
241   translation 0.25 1.75 1.75
242   children [
243     USE Atom
244   ]
245 }
```

```
246 DEF MoveIt Script {
247   directOutput TRUE
248   url "particle.class"
249   eventIn   SFFloat  clicked
250   field     SFNode   node1   USE ATOM1
251   field     SFNode   node2   USE ATOM2
252   field     SFNode   node3   USE ATOM3
253   field     SFNode   node4   USE ATOM4
254   field     SFNode   node5   USE ATOM5
255   field     SFNode   node6   USE ATOM6
256   field     SFNode   node7   USE ATOM7
257   field     SFNode   node8   USE ATOM8
258   field     SFNode   node9   USE ATOM9
259   field     SFNode   node10   USE ATOM10
260   field     SFNode   node11   USE ATOM11
261   field     SFNode   node12   USE ATOM12
262   field     SFNode   node13   USE ATOM13
263   field     SFNode   node14   USE ATOM14
264   field     SFNode   node15   USE ATOM15
265   field     SFNode   node16   USE ATOM16
266   field     SFNode   node17   USE ATOM17
267   field     SFNode   node18   USE ATOM18
268   field     SFNode   node19   USE ATOM19
269   field     SFNode   node20   USE ATOM20
270   field     SFNode   node21   USE ATOM21
271   field     SFNode   node22   USE ATOM22
272   field     SFNode   node23   USE ATOM23
273   field     SFNode   node24   USE ATOM24
274   field     SFNode   node25   USE ATOM25
275   field     SFNode   node26   USE ATOM26
276   field     SFNode   node27   USE ATOM27
277   field     SFNode   node28   USE ATOM28
278   field     SFNode   node29   USE ATOM29
279   field     SFNode   node30   USE ATOM30
280   field     SFNode   node31   USE ATOM31
281   field     SFNode   node32   USE ATOM32
282   field     SFBool   on FALSE
283 }
284 DEF MyTimer TimeSensor {
285   loop TRUE
286   cycleInterval 2
287 }
288 ROUTE MyTimer.fraction_changed TO MoveIt.clicked
```

図16－2　シミュレーション教材のVRMLファイル（その2）

係している。vel( ) (203から209行目)において，最終的な単位時間ステップ経過後の速度を計算している。outvrml( ) (300から333行目)では，求められた粒子の座標を，そのままバーチャルリアリティ空間内の粒子の座標の更新値となるように出力している。なお，このプログラムでは，mov( )，force( )，vel( ) の計算をループ処理で10回繰り返した（10ステップ）後に，粒子座標の更新を行うことにしている。そして，その繰り返し中は温度の調整は行わない。10ステップごとにプログラム全体が繰り返し呼び出されて実行されるが，その呼び出し時にのみ温度調整を行う。

4．まとめと展開

本研究において作成した教材の一つは，学部建造物のバーチャルリアリティ空間内に，研究紹介として科学技術教育用シミュレーションを取り入れた内容を有していた。このバーチャルリアリティの構築については，現時点では，まだ全体を完成するところにまで至っていないので，さらに追加・改良する点が残されている。今後も本研究結果を踏まえて，さらにバーチャルリアリティとしての内容を充実したものにして行く必要がある。そして，最終的には，ある特定の学部のみならず，大学全体のバーチャルリアリティを構築することを目指している。VRMLによるバーチャルリアリティは，インターネット用のブラウザソフトを用いて見ることができる。そのため，今回作成したような学部や研究室紹介およびシミュレーション等のバーチャルリアリティを，大学や学部のホームページ等の中に取り入れてインターネット上で公開するで，学内外を問わず，すべての見る者に対して興

```
import vrml.*;
import vrml.field.*;
import vrml.node.*;
//
public class particle extends Script {
  int N=32;
  double lleng=2.0;
  double hleng=lleng/2.0;
  double dt=0.03;
  double dtsh=dt*dt*0.5;
  double tref=0.100;
   double sigm=0.171;
  double efx=0.3, efy=0.0, efz=0.0;
//
  double fx []=new double[33]; double fy []=new double[33]; double fz []=new double[33];
  double x []=new double[33];  double y []=new double[33];  double z []=new double[33];
  double u []=new double[33];  double v []=new double[33];  double w []=new double[33];
//-------------------------------------------------
  private SFNode theNode1 ;   private SFNode theNode2 ;   private SFNode theNode3 ;
  private SFNode theNode4 ;   private SFNode theNode5 ;   private SFNode theNode6 ;
  private SFNode theNode7 ;   private SFNode theNode8 ;   private SFNode theNode9 ;
  private SFNode theNode10 ;  private SFNode theNode11 ;  private SFNode theNode12 ;
  private SFNode theNode13 ;  private SFNode theNode14 ;  private SFNode theNode15 ;
  private SFNode theNode16 ;  private SFNode theNode17 ;  private SFNode theNode18 ;
  private SFNode theNode19 ;  private SFNode theNode20 ;  private SFNode theNode21 ;
  private SFNode theNode22 ;  private SFNode theNode23 ;  private SFNode theNode24 ;
  private SFNode theNode25 ;  private SFNode theNode26 ;  private SFNode theNode27 ;
  private SFNode theNode28 ;  private SFNode theNode29 ;  private SFNode theNode30 ;
  private SFNode theNode31 ;  private SFNode theNode32 ;
  private SFVec3f position1;   private SFVec3f position2;   private SFVec3f position3;
  private SFVec3f position4;   private SFVec3f position5;   private SFVec3f position6;
  private SFVec3f position7;   private SFVec3f position8;   private SFVec3f position9;
  private SFVec3f position10;  private SFVec3f position11;  private SFVec3f position12;
  private SFVec3f position13;  private SFVec3f position14;  private SFVec3f position15;
  private SFVec3f position16;  private SFVec3f position17;  private SFVec3f position18;
  private SFVec3f position19;  private SFVec3f position20;  private SFVec3f position21;
  private SFVec3f position22;  private SFVec3f position23;  private SFVec3f position24;
  private SFVec3f position25;  private SFVec3f position26;  private SFVec3f position27;
  private SFVec3f position28;  private SFVec3f position29;  private SFVec3f position30;
  private SFVec3f position31;  private SFVec3f position32;
//=================================================================
 public void processEvent(Event e) {
    inptvrml();
    force();
    vscale();
   for(int il=1; il<=10; il++){
    mov();
    force();
    vel();
   }
    outvrml();
}
//=================================================================
public void initialize() {
    theNode1 = (SFNode) getField("node1");    theNode2 = (SFNode) getField("node2");
    theNode3 = (SFNode) getField("node3");    theNode4 = (SFNode) getField("node4");
    theNode5 = (SFNode) getField("node5");    theNode6 = (SFNode) getField("node6");
    theNode7 = (SFNode) getField("node7");    theNode8 = (SFNode) getField("node8");
    theNode9 = (SFNode) getField("node9");    theNode10 = (SFNode) getField("node10");
    theNode11 = (SFNode) getField("node11");  theNode12 = (SFNode) getField("node12");
    theNode13 = (SFNode) getField("node13");  theNode14 = (SFNode) getField("node14");
    theNode15 = (SFNode) getField("node15");  theNode16 = (SFNode) getField("node16");
    theNode17 = (SFNode) getField("node17");  theNode18 = (SFNode) getField("node18");
    theNode19 = (SFNode) getField("node19");  theNode20 = (SFNode) getField("node20");
    theNode21 = (SFNode) getField("node21");  theNode22 = (SFNode) getField("node22");
    theNode23 = (SFNode) getField("node23");  theNode24 = (SFNode) getField("node24");
    theNode25 = (SFNode) getField("node25");  theNode26 = (SFNode) getField("node26");
    theNode27 = (SFNode) getField("node27");  theNode28 = (SFNode) getField("node28");
    theNode29 = (SFNode) getField("node29");  theNode30 = (SFNode) getField("node30");
    theNode31 = (SFNode) getField("node31");  theNode32 = (SFNode) getField("node32");
 }
//=================================================================
void vscale(){
  double tscale=16.0/(1.0*N-1.0);
  double u1=0.0,u2=0.0,v1=0.0,v2=0.0,w1=0.0,w2=0.0;
  double su,sv,sw;
  double ru,rv,rw;
  double avu,avv,avw;
  double ekin;
  double ts,sc;
//
  for (int i=1; i<=N; i=i+2){
     su=5.0;
    while(su>=1.0){
     u1=2.0*Math.random()-1.0;
     u2=2.0*Math.random()-1.0;
     su=u1*u1+u2*u2;
    }
     sv=5.0;
    while(sv>=1.0){
     v1=2.0*Math.random()-1.0;
     v2=2.0*Math.random()-1.0;
     sv=v1*v1+v2*v2;
    }
     sw=5.0;
    while(sw>=1.0){
     w1=2.0*Math.random()-1.0;
     w2=2.0*Math.random()-1.0;
     sw=w1*w1+w2*w2;
```

図17－1　シミュレーション教材の Java プログラム（その１）

```
100     }
101     ru=-2.0*Math.log(su)/su;
102     rv=-2.0*Math.log(sv)/sv;
103     rw=-2.0*Math.log(sw)/sw;
104
105     u[i] = u1*Math.sqrt(ru);
106     u[i+1]=u2*Math.sqrt(ru);
107     v[i] = v1*Math.sqrt(rv);
108     v[i+1]=v2*Math.sqrt(rv);
109     w[i] = w1*Math.sqrt(rw);
110     w[i+1]=w2*Math.sqrt(rw);
111   }
112
113   avu=0.0;
114   avv=0.0;
115   avw=0.0;
116   for (int i=1; i<=N; i++){
117    avu=avu+u[i];
118    avv=avv+v[i];
119    avw=avw+w[i];
120   }
121   avu=avu/N;
122   avv=avv/N;
123   avw=avw/N;
124   ekin=0.0;
125   for (int i=1; i<=N; i++){
126    u[i]=u[i]-avu;
127    v[i]=v[i]-avv;
128    w[i]=w[i]-avw;
129    ekin=ekin+u[i]*u[i]+v[i]*v[i]+w[i]*w[i];
130   }
131
132   ts=tscale*ekin;
133   sc=dt*Math.sqrt(tref/ts);
134   for (int i=1; i<=N; i++){
135    u[i]=u[i]*sc;
136    v[i]=v[i]*sc;
137    w[i]=w[i]*sc;
138   }
139 }
140 //=============================================================================
141 void mov(){
142   for(int i=1;i<=N;i++){
143     x[i]=x[i]+u[i]+fx[i];
144     y[i]=y[i]+v[i]+fy[i];
145     z[i]=z[i]+w[i]+fz[i];
146       x[i]=x[i]-lleng*Math.floor(x[i]/lleng);
147       y[i]=y[i]-lleng*Math.floor(y[i]/lleng);
148       z[i]=z[i]-lleng*Math.floor(z[i]/lleng);
149   }
150   for(int i=1;i<=N;i++){
151    u[i]=u[i]+fx[i];
152    v[i]=v[i]+fy[i];
153    w[i]=w[i]+fz[i];
154   }
155 }
156 //=============================================================================
157 void force(){
158   double fxtmp,fytmp,fztmp;
159   double rc=2.0;
160   double rx,ry,rz,rabs;
161   double sdr6,sdr12,sdrt,us;
162   int ip1;
163 //
164   for (int ii=1;ii<=N;ii++){
165     fx[ii]=0.0;
166     fy[ii]=0.0;
167     fz[ii]=0.0;
168   }
169 //
170  for(int i=1;i<=N;i++){
171    fxtmp=0.0;
172    fytmp=0.0;
173    fztmp=0.0;
174    ip1=i+1;
175   for(int j=ip1;j<=N;j++){
176
177       rx=x[i]-x[j]-lleng*Math.rint((x[i]-x[j])/lleng);
178       ry=y[i]-y[j]-lleng*Math.rint((y[i]-y[j])/lleng);
179       rz=z[i]-z[j]-lleng*Math.rint((z[i]-z[j])/lleng);
180
181     rabs=Math.sqrt(rx*rx+ry*ry+rz*rz);
182     us=sigm*sigm*sigm/(rabs*rabs*rabs);
183       fxtmp=fxtmp+us*rx;
184       fytmp=fytmp+us*ry;
185       fztmp=fztmp+us*rz;
186       fx[j]=fx[j]-us*rx;
187       fy[j]=fy[j]-us*ry;
188       fz[j]=fz[j]-us*rz;
189 //     }
190   }
191     fx[i]=fx[i]+fxtmp +efx ;
192     fy[i]=fy[i]+fytmp +efy ;
193     fz[i]=fz[i]+fztmp +efz ;
194 }
195 //
196  for (int l=1;l<=N;l++){
197     fx[l]=fx[l]*dtsh;
198     fy[l]=fy[l]*dtsh;
```

図17－2　シミュレーション教材のJavaプログラム（その2）

```
199     fz[l]=fz[l]*dtsh;
200   }
201 }
202 //=============================================================================
203 void vel(){
204   for(int i=1;i<=N;i++){
205    u[i]=u[i]+fx[i];
206    v[i]=v[i]+fy[i];
207    w[i]=w[i]+fz[i];
208   }
209 }
210 //=============================================================================
211 void inptvrml() {
212    Node node1,node2,node3,node4,node5,node6,node7,node8,node9;
213    Node node10,node11,node12,node13,node14,node15,node16,node17,node18;
214    Node node19,node20,node21,node22,node23,node24,node25,node26,node27;
215    Node node28,node29,node30,node31,node32;
216 //
217       node1 = (Node)(theNode1.getValue());    node2 = (Node)(theNode2.getValue());
218       node3 = (Node)(theNode3.getValue());    node4 = (Node)(theNode4.getValue());
219       node5 = (Node)(theNode5.getValue());    node6 = (Node)(theNode6.getValue());
220       node7 = (Node)(theNode7.getValue());    node8 = (Node)(theNode8.getValue());
221       node9 = (Node)(theNode9.getValue());    node10 = (Node)(theNode10.getValue());
222       node11 = (Node)(theNode11.getValue());  node12 = (Node)(theNode12.getValue());
223       node13 = (Node)(theNode13.getValue());  node14 = (Node)(theNode14.getValue());
224       node15 = (Node)(theNode15.getValue());  node16 = (Node)(theNode16.getValue());
225       node17 = (Node)(theNode17.getValue());  node18 = (Node)(theNode18.getValue());
226       node19 = (Node)(theNode19.getValue());  node20 = (Node)(theNode20.getValue());
227       node21 = (Node)(theNode21.getValue());  node22 = (Node)(theNode22.getValue());
228       node23 = (Node)(theNode23.getValue());  node24 = (Node)(theNode24.getValue());
229       node25 = (Node)(theNode25.getValue());  node26 = (Node)(theNode26.getValue());
230       node27 = (Node)(theNode27.getValue());  node28 = (Node)(theNode28.getValue());
231       node29 = (Node)(theNode29.getValue());  node30 = (Node)(theNode30.getValue());
232       node31 = (Node)(theNode31.getValue());  node32 = (Node)(theNode32.getValue());
233       position1 = (SFVec3f)(node1.getExposedField("translation"));
234       position2 = (SFVec3f)(node2.getExposedField("translation"));
235       position3 = (SFVec3f)(node3.getExposedField("translation"));
236       position4 = (SFVec3f)(node4.getExposedField("translation"));
237       position5 = (SFVec3f)(node5.getExposedField("translation"));
238       position6 = (SFVec3f)(node6.getExposedField("translation"));
239       position7 = (SFVec3f)(node7.getExposedField("translation"));
240       position8 = (SFVec3f)(node8.getExposedField("translation"));
241       position9 = (SFVec3f)(node9.getExposedField("translation"));
242       position10 = (SFVec3f)(node10.getExposedField("translation"));
243       position11 = (SFVec3f)(node11.getExposedField("translation"));
244       position12 = (SFVec3f)(node12.getExposedField("translation"));
245       position13 = (SFVec3f)(node13.getExposedField("translation"));
246       position14 = (SFVec3f)(node14.getExposedField("translation"));
247       position15 = (SFVec3f)(node15.getExposedField("translation"));
248       position16 = (SFVec3f)(node16.getExposedField("translation"));
249       position17 = (SFVec3f)(node17.getExposedField("translation"));
250       position18 = (SFVec3f)(node18.getExposedField("translation"));
251       position19 = (SFVec3f)(node19.getExposedField("translation"));
252       position20 = (SFVec3f)(node20.getExposedField("translation"));
253       position21 = (SFVec3f)(node21.getExposedField("translation"));
254       position22 = (SFVec3f)(node22.getExposedField("translation"));
255       position23 = (SFVec3f)(node23.getExposedField("translation"));
256       position24 = (SFVec3f)(node24.getExposedField("translation"));
257       position25 = (SFVec3f)(node25.getExposedField("translation"));
258       position26 = (SFVec3f)(node26.getExposedField("translation"));
259       position27 = (SFVec3f)(node27.getExposedField("translation"));
260       position28 = (SFVec3f)(node28.getExposedField("translation"));
261       position29 = (SFVec3f)(node29.getExposedField("translation"));
262       position30 = (SFVec3f)(node30.getExposedField("translation"));
263       position31 = (SFVec3f)(node31.getExposedField("translation"));
264       position32 = (SFVec3f)(node32.getExposedField("translation"));
265 //
266       x[1]=position1.getX();  y[1]=position1.getY();  z[1]=position1.getZ();
267       x[2]=position2.getX();  y[2]=position2.getY();  z[2]=position2.getZ();
268       x[3]=position3.getX();  y[3]=position3.getY();  z[3]=position3.getZ();
269       x[4]=position4.getX();  y[4]=position4.getY();  z[4]=position4.getZ();
270       x[5]=position5.getX();  y[5]=position5.getY();  z[5]=position5.getZ();
271       x[6]=position6.getX();  y[6]=position6.getY();  z[6]=position6.getZ();
272       x[7]=position7.getX();  y[7]=position7.getY();  z[7]=position7.getZ();
273       x[8]=position8.getX();  y[8]=position8.getY();  z[8]=position8.getZ();
274       x[9]=position9.getX();  y[9]=position9.getY();  z[9]=position9.getZ();
275       x[10]=position10.getX();y[10]=position10.getY();z[10]=position10.getZ();
276       x[11]=position11.getX();y[11]=position11.getY();z[11]=position11.getZ();
277       x[12]=position12.getX();y[12]=position12.getY();z[12]=position12.getZ();
278       x[13]=position13.getX();y[13]=position13.getY();z[13]=position13.getZ();
279       x[14]=position14.getX();y[14]=position14.getY();z[14]=position14.getZ();
280       x[15]=position15.getX();y[15]=position15.getY();z[15]=position15.getZ();
281       x[16]=position16.getX();y[16]=position16.getY();z[16]=position16.getZ();
282       x[17]=position17.getX();y[17]=position17.getY();z[17]=position17.getZ();
283       x[18]=position18.getX();y[18]=position18.getY();z[18]=position18.getZ();
284       x[19]=position19.getX();y[19]=position19.getY();z[19]=position19.getZ();
285       x[20]=position20.getX();y[20]=position20.getY();z[20]=position20.getZ();
286       x[21]=position21.getX();y[21]=position21.getY();z[21]=position21.getZ();
287       x[22]=position22.getX();y[22]=position22.getY();z[22]=position22.getZ();
288       x[23]=position23.getX();y[23]=position23.getY();z[23]=position23.getZ();
289       x[24]=position24.getX();y[24]=position24.getY();z[24]=position24.getZ();
290       x[25]=position25.getX();y[25]=position25.getY();z[25]=position25.getZ();
291       x[26]=position26.getX();y[26]=position26.getY();z[26]=position26.getZ();
292       x[27]=position27.getX();y[27]=position27.getY();z[27]=position27.getZ();
293       x[28]=position28.getX();y[28]=position28.getY();z[28]=position28.getZ();
294       x[29]=position29.getX();y[29]=position29.getY();z[29]=position29.getZ();
295       x[30]=position30.getX();y[30]=position30.getY();z[30]=position30.getZ();
296       x[31]=position31.getX();y[31]=position31.getY();z[31]=position31.getZ();
297       x[32]=position32.getX();y[32]=position32.getY();z[32]=position32.getZ();
```

図17－3　シミュレーション教材の Java プログラム（その3）

```
298 }
299 //=============================================================================
300 void outvrml() {
301     position1.setValue( (float)x[1],  (float)y[1],  (float)z[1]);
302     position2.setValue( (float)x[2],  (float)y[2],  (float)z[2]);
303     position3.setValue( (float)x[3],  (float)y[3],  (float)z[3]);
304     position4.setValue( (float)x[4],  (float)y[4],  (float)z[4]);
305     position5.setValue( (float)x[5],  (float)y[5],  (float)z[5]);
306     position6.setValue( (float)x[6],  (float)y[6],  (float)z[6]);
307     position7.setValue( (float)x[7],  (float)y[7],  (float)z[7]);
308     position8.setValue( (float)x[8],  (float)y[8],  (float)z[8]);
309     position9.setValue( (float)x[9],  (float)y[9],  (float)z[9]);
310     position10.setValue((float)x[10], (float)y[10], (float)z[10]);
311     position11.setValue((float)x[11], (float)y[11], (float)z[11]);
312     position12.setValue((float)x[12], (float)y[12], (float)z[12]);
313     position13.setValue((float)x[13], (float)y[13], (float)z[13]);
314     position14.setValue((float)x[14], (float)y[14], (float)z[14]);
315     position15.setValue((float)x[15], (float)y[15], (float)z[15]);
316     position16.setValue((float)x[16], (float)y[16], (float)z[16]);
317     position17.setValue((float)x[17], (float)y[17], (float)z[17]);
318     position18.setValue((float)x[18], (float)y[18], (float)z[18]);
319     position19.setValue((float)x[19], (float)y[19], (float)z[19]);
320     position20.setValue((float)x[20], (float)y[20], (float)z[20]);
321     position21.setValue((float)x[21], (float)y[21], (float)z[21]);
322     position22.setValue((float)x[22], (float)y[22], (float)z[22]);
323     position23.setValue((float)x[23], (float)y[23], (float)z[23]);
324     position24.setValue((float)x[24], (float)y[24], (float)z[24]);
325     position25.setValue((float)x[25], (float)y[25], (float)z[25]);
326     position26.setValue((float)x[26], (float)y[26], (float)z[26]);
327     position27.setValue((float)x[27], (float)y[27], (float)z[27]);
328     position28.setValue((float)x[28], (float)y[28], (float)z[28]);
329     position29.setValue((float)x[29], (float)y[29], (float)z[29]);
330     position30.setValue((float)x[30], (float)y[30], (float)z[30]);
331     position31.setValue((float)x[31], (float)y[31], (float)z[31]);
332     position32.setValue((float)x[32], (float)y[32], (float)z[32]);
333 }
334 //=============================================================================
335 }
336
337
```

図17－4　シミュレーション教材のJavaプログラム（その4）

味・関心を抱かせることが可能になるであろう。そして，このような教材についての，さらなる研究展開の方向として，マルチメディア教材への応用が考えられる。つまり，VRML(バージョン2.0)では，バーチャルリアリティ空間内においてマルチメディアデータを扱える。例えば，ビデオ撮影した画像・サウンドをＭＰＥＧ形式データとして，サウンドのみの場合はWAVまたはMIDI形式のデータとして用意することで，動画やサウンドをバーチャルリアリティ空間内で表示・再生できる。このような教材の開発は，多様化した情報関連技術を取り入れて活用してゆく必要性のある情報教育分野において，ますます重要な課題となるものと思われる。

さらに本研究では，別のバーチャルリアリティの利用の仕方として，荷電粒子系の運動をリアルタイムで3次元可視化することが可能な科学技術教育用シミュレーション教材の開発も行った。その中で，高価な若しくは複雑な専用可視化ソフトウェアを用いるのではなく，インターネットに関連したVRMLのブラウザで，3次元可視化されたシミュレーション教材が，非常に安価に簡単に作成されることを示した。これは，実際の教育現場での教材利用を考えた場合，重要であり強調されるべき事である。現在の著しいインターネット関連の技術開発の中で作り出された各種ブラウザの特徴を捉え，その機能を利用して教材を開発することは，これからの科学技術教育において非常に有効な手段となり得るであろう。

謝辞

VRMLによる学部建造物のバーチャルリアリティ作成段階における柴田真之氏の協力に感謝します。

文献

(1) 一つの試みとして次を参照．川口高明：応用物理教育，1998，Vol.22，Ｐ．3.
(2) VRMLについては，例えば次を参照。Rodger Lea，松田晃一，宮下健：VRML＋Java，プレンティスホール，1997．J.Hartman and J.Wernecke：VRML2.0ハンドブック，アジソン・ウェスレイ，1997．中山茂：VRML2，技報堂出版，1997．河西朝雄，河西雄一：VRML2.0，ナツメ社，1998.
(3) 川口高明：応用物理教育，1997，Vol.21，Ｐ．45.
(4) Javascriptについては，例えば次を参照．上田学：JavaScript，オーム社，1997．松尾忠則，古籏一浩：JavaScript，インプレス，1997．大津真：JavaScriptパーフェクトリファレンス，BNN出版，1997.

(5) C.Jacoboni and P.Lugli:The Monte Carlo Method for Semiconductor Device Simulation, Springer-Verlag, 1989.
(6) D.W.Heermann:Computer Simulation Methods in Theoretical Physics(2nd edn.), Springer-Verlag, 1990.
(7) 上田顕：コンピュータシミュレーション,朝倉書店, 1990.
(8) 津田孝夫：モンテカルロ法とシミュレーション，培風館，1997.
(9) 伏見正則：確率的方法とシミュレーション，岩波書店，1994.
(10) SONY製VRMLブラウザ(Community Place Browser)およびそのプラグイン版については，Webサイト（http://vs.sony.co.jp/）を参照.
(11) 小竹進：分子熱流体，丸善，1990.