Distortion-oriented 技術による3次元シーンの自動簡略化手法

井上正行[†,☆] 小池英樹[†]

仮想現実感システムや情報視覚化システムに代表される対話的3次元グラフィックスシステムにおいては大量のポリゴンを表示しつつも，対話性を確保することが重要な問題である．本論文ではdistortion-oriented技術を用いることによって3次元シーンを自動的に簡略化する手法について述べた．具体的には，ユーザイベントがない場合，十分な描画時間があるのですべてのオブジェクトを表示する．しかし，ユーザイベントが生じた際には，着目しているオブジェクトとその近傍のオブジェクトのみを表示し，他は描画しない．我々は，この近傍を決定するためにdistortion-orientedアルゴリズムの1つであるFractalViewsを用いた．本手法は仮想空間ウォークスルー，3次元CADデータ，分子モデル視覚化に適用され，その効果を確認した．

Automatic Simplification of the 3-D Scene Using Distortion-oriented Approach

MASAYUKI INOUE[†,☆] and HIDEKI KOIKE[†]

In interactive 3-D graphics systems, such as virtual reality systems, information visualization systems, etc., it is important to manipulate a large number of polygons and to maintain high interactivity. This paper proposed a technique to simplify the 3-D scene automatically by using distortion-oriented algorithm. While no user event occurs, the system renders all objects in the scene graph since there is enough time to do so. When a user event is detected, the system renders a current focused object and its neighborhood (in the scene graph). To decide the neighborhood, FractalViews, which is a variation of distortion-oriented algorithms, is used. The technique was applied to virtual-walkthrough, 3-D CAD data, and molecular visualization.

1. はじめに

1.1 背景

近年におけるグラフィックスハードウェアの高性能化，低価格化にともない，より複雑でより現実感の高い仮想空間を比較的容易に作成することが可能になった．こうした現実感の高い仮想空間は，たとえばテクスチャマッピングを用いたり，オブジェクトを構成するポリゴンの数を増やすなどして生成される．

仮想現実感システム，情報視覚化システム，CADシステム，サイエンティフィックビジュアリゼーションシステムといった対話的3次元コンピュータグラフィックス（CG）システムにおいては，いかにスムーズに空間内で視点移動を行うか，いかに対話的にオブジェクトの操作を行うことができるかが重要な問題である．しかし，仮想空間に現実感を追求すればするほど，モデルを複雑にする必要がある．また，大規模なデータを視覚化しようとすると描画すべきポリゴン数が増加する．結果としてシステムにかかる負荷が増大し，対話的操作が困難になるというジレンマが生じる．

複雑にレンダリングされたオブジェクトが存在する3次元空間では，システムの対話性を向上させるために，ある程度描画の質を落として（ポリゴン数を減らしたり，テクスチャマッピングを行わずに）画面に表示する必要がある．すなわち，複雑で現実感の高い仮想空間を追求することと，対話性を確保することはトレードオフの関係にある．

1.2 関連研究

この問題は現在，Time-Critical Rendering（TCR）の名のもとに研究が行われている．代表的な例はVRML[11]等のLevels of Detail（LOD）機能である．

† 電気通信大学大学院情報システム学研究科
Graduate School of Information Systems, University of Electro-Communications
☆ 現在，セガ・エンタープライゼス
Presently with SEGA Enterprises, Ltd.

VRMLにおけるLODは1つのオブジェクトに対して詳細度の異なる複数のモデルを定義しておき，観察者からの距離に応じて描画するモデルを切り替える．この手法は実現が比較的簡単だが，各フレームの描画時間は考慮されていない．

フレーム描画時間の均一化を重視した研究にはFunkhouserら[4]の手法がある．この手法はVRML同様，各オブジェクトに対して詳細度の異なる複数のモデルを準備する．そして観察者からの距離だけでなく，各オブジェクトが描画されたときの視野における大きさをも考慮に入れ，描画にかかると思われる時間を動的に計算する．そして，フレーム描画時間が均一になるように，各オブジェクトごとにどのモデルを描画するかを決定する．

これらの手法に共通する問題点は，1つのオブジェクトに対し複数のモデルを記述しなければならない点である．モデル作成にかかるコストよりもその品質が重視されるような3次元アプリケーションの場合には，こうした手法は有効である．これに対し，一般の人々が趣味で作成するVRMLアプリケーション等の場合にはモデル作成の簡単さが重要であると思われる．また，大規模な情報視覚化システムでは，個々のデータは比較的簡単な形状で表現されるため，こうした複数のモデルを定義する手法は有効でない．

本論文では，複数のモデルを定義することなしに対話性を確保する手法について述べた．我々は，distortion-oriented技術と呼ばれている手法[10]を用いることによって複雑なシーンの自動簡略化を行った．具体的には，ユーザが3次元空間で視点移動やオブジェクトの操作を行っていないときには空間内のすべてのオブジェクトの描画を行う．一方，ユーザイベントが発生したときには着目しているオブジェクトとその“近傍”のオブジェクトのみを描画する．近傍の決定には後述するFractalViewsを用いた．以下，次章では我々の手法の基本的部分を説明する．3章では本手法の実装について述べ，4章で仮想空間ウォークスルーへの応用を示す．5章では我々の手法を用いて実現できる拡張的機能について説明する．6章では，3次元CADデータと分子モデル視覚化への応用を示す．7章で考察を行い，8章でまとめる．

2. 基本的戦略

2.1 シーングラフ

3次元空間の構成要素，形，色，座標，テキスト等を階層的に表現したグラフ構造のことをシーングラフという．シーングラフはOpen Inventor[15]やVRML等で用いられているデータ構造であり，シーンの描画はこのシーングラフを深さ優先順に走査することによって行われる．簡単なシーングラフの例を図1に示す．

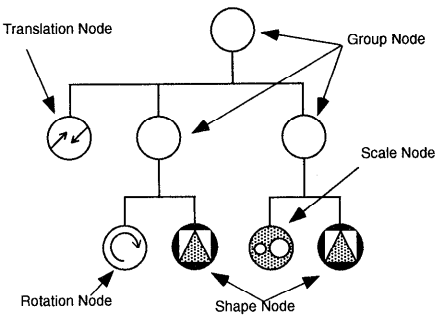


図1 シーングラフの例
Fig. 1 An example scene graph.

2.2 Distortion-oriented技術とFractalViews

Distortion-oriented技術[10]とは，着目している情報（オブジェクト）は詳しく見せつつ，大域的な情報も表示する技術のことである．たとえばFurnas[5]のfisheye viewsは，多くの情報視覚化システムで用いられている[2),5),14)]．

我々は，distortion-orientedアルゴリズムの1つであるFractalViews[8]と呼ばれるアルゴリズムを採用した．FractalViewsは，フラクタルの概念を用いた提示情報量制御手法の1つである．この手法は木として表現できる情報構造に対して適用可能であり，その特徴として，

(1) 着目点（ノード）とその近傍を着目点の変更に関係なくほぼ一定の数だけ提示できる
(2) 情報の提示量を柔軟に設定できる

の2点があげられる．

FractalViewsアルゴリズムを簡単に説明する．まず，対象とする情報構造において現在の着目点をルートとする木を定義する．次にこの木に対し，各ノードにおける分岐数をN_x，そこでのスケール・ファクタをr_xとしたとき，各ノードのフラクタル値Fvを以下の式に従って決定する．

$$\begin{cases} Fv_{root} = 1 \\ Fv_{child\_of\_x} = r_x Fv_x \end{cases} \tag{1}$$

ただし

$$r_x = CN_x^{-1/D}. \tag{2}$$

Dは適当な定数，Cは$0 < C \leq 1$の定数である．

ここで$D = -1$，$C = 1$とすることで，式(1)は

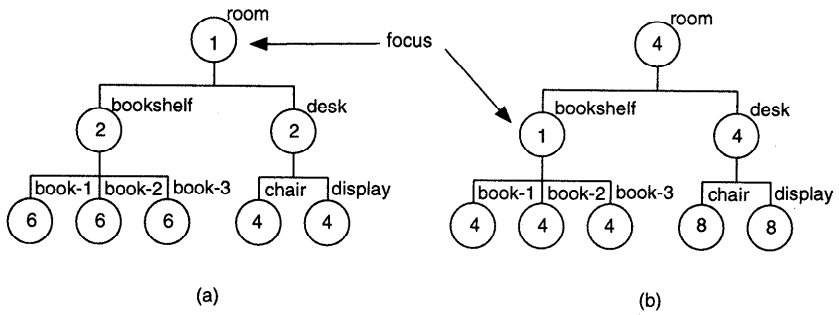


図 2 フラクタル値の伝播例. (a) 着目点は "room", (b) 着目点は "bookshelf"
Fig. 2 An example propagation. (a) The focus is on "room", (b) The focus is on "bookshelf".

$$\begin{cases} Fv_{root} = 1 \\ Fv_{child\_of\_x} = N_x Fv_x \end{cases} \quad (3)$$

と簡略化できる. 伝播される値は木の末端へ行くほど大きくなり, また分岐が多いときほど子ノードに伝播される値は大きくなる. 任意の閾値 k を選び, それ以下の値を持つノードを表示することにすると, 閾値に従い異なる表示を得ることができる. 伝播の具体例は次節で示す.

2.3 シーングラフ + FractalViews

我々は, FractalViews をシーングラフに適用することによって, 3 次元空間内でのシーンの簡略化を実現した. 具体的には, シーングラフ上でユーザの着目するオブジェクトから式 (3) に従って各オブジェクトのフラクタル値を計算する. そして, ユーザの設定する閾値より小さい値を持つオブジェクトは表示し, 他は表示しない.

フラクタル値伝播の例を図 **2** に示す. 本来, シーングラフは図 1 のように Shape ノードや Translation ノード等が存在する. しかし図 2 ではそれらを省略した形で示している. なぜなら, 子を持つことができるノードは Group ノードのみなので, フラクタル値を持つことができるのも Group ノードのみにしたからである. したがって, フラクタル値の計算を行う際にも Group ノードの親子関係のみに着目して計算を行う.

図 2 (a) では着目点を "room" に設定してある. したがって, このノードのフラクタル値は 1 である. "room" は 2 つの子ノードを持つので自分のフラクタル値 1 を子ノードの数 2 で掛けた値を各子ノードに伝播させる. 子ノードは自分のフラクタル値を用いてさらに孫ノードへとフラクタル値の伝播を行う. ここで閾値を 4 に設定すると, "book-1", "book-2", "book-3" は表示されない.

```
#include "fvnode.h"

int main(int argc, char **argv) {
  // ツールキットの初期化
  fvinit(argc, argv, 500, 500);

  // ルートノードの作成
  fvGroup *root = new fvGroup();

  // 透視投影カメラの定義とルートノードへの登録
  PerspectiveCamera *cam =
        new PerspectiveCamera;
  cam->setPosition(5.0, 5.0, 5.0);
  root->addChold(cam);

  // 点光源の定義とルートノードへの登録
  DirectionalLight *lt = new DirectionalLight;
  root->addChild(lt);

  // 立方体の定義とルートノードへの登録
  Cube *c = new Cube;
  root->addChild(c);

  // root をシーングラフのルートとして登録
  fvSetWorld(root);
  // 視点移動用関数へカメラを登録
  fvSetCamera(cam);
  // メインループ
  fvMainLoop();
}
```

図 3 ツールキットを用いたプログラム例
Fig. 3 An example program using our toolkit.

次に, 図 2 (b) では着目点を "bookshelf" に移動した. システムは "bookshelf" を新たなフラクタル値伝播のルートとしてフラクタル値の再計算を行う (注意: シーングラフの木とフラクタル値伝播の木は異なる). "bookshelf" はシーングラフ上での 3 つの子ノードと 1 つの親ノード, つまり 4 分岐を持つので, これら子ノードのフラクタル値は 4 となる. 以下同様に計算を

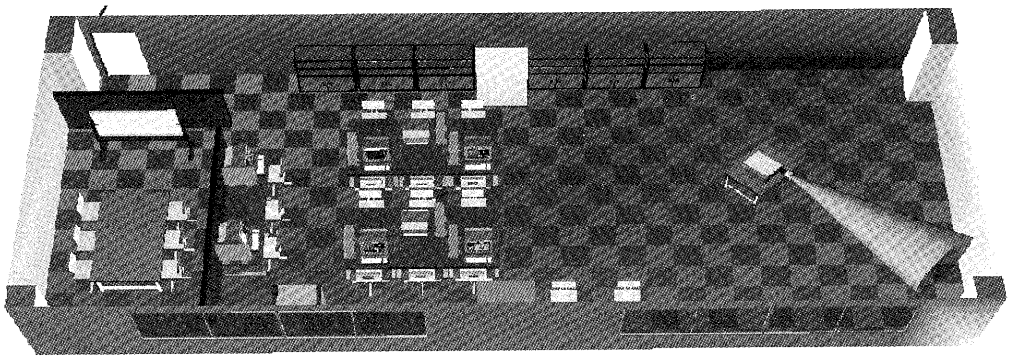


図4 仮想空間全体図
Fig. 4 An overview of the room used in virtual walkthrough demonstration.

行う.

3. 実　　装

我々は，前章で述べた手法をグラフィックスツールキットとして実装した．ツールキットは現在，Silicon Graphics 社（SGI）のワークステーション上で動作している．本ツールキットは，通常では記述が複雑となりがちなオブジェクト管理，座標変換，カメラ，ライトの設定といった部分を C++ 上のクラスライブラリというかたちでカプセルコーディング化し，実装を行った．実行時には内部で OpenGL を呼ぶ．本ツールキットは Inventor を意識して作成しており，3次元空間を作成する際には Inventor と同じようにプログラミングできるようにした．簡単なプログラム例を**図3** に示す．なお，後述する性能評価は SGI Indigo2 IMPACT（R4400/200 MHz, 96 MB メモリ）を用いた．

4. 応用例1：バーチャルウォークスルー

本ツールキットを用いて3次元仮想空間のウォークスルーのデモンストレーションを作成した．作成した部屋を**図4** に示す．

シーングラフの作成にあたり，この例ではまず部屋を左，右，中央に分割し，この3つの子供を部屋（root ノード）の子供とした．部屋の中央は，多少込み入っているので，部屋の中央の部分だけ，さらに，前，後ろ，中央と3つに分割した（**図5**）．このように分割した領域に対してさまざまなオブジェクトの配置を行った．オブジェクトもまた階層構造になるようにした．たとえば，机の子供として椅子やディスプレイが設定されている．

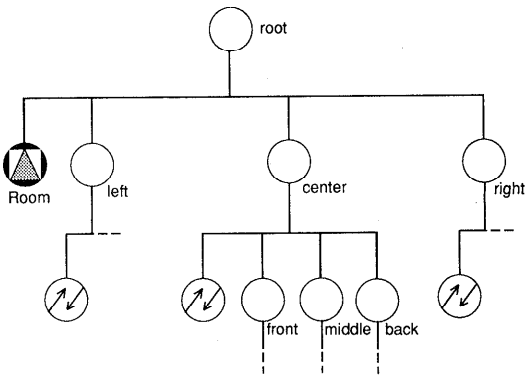


図5 部屋のシーングラフ
Fig. 5 A scene graph for the room.

4.1 結　　果

結果の画面を**図6** に示す．この例では着目点を画面中央の机に設定している．図6(a) は全部のオブジェクトを描画した場合であり，視点移動が行われていないときの画面がこれにあたる．この場合1フレームを描画するのに必要な時間は約 0.422 秒，つまりフレームレートは 2.37（frame/sec）であった．図6(b) は閾値を小さくとった場合であり，視点移動が生じた際に描画される画面がこれになる．フレームレートは 71.43 であるが，図6(a) の画面に比べると描画されていないオブジェクトが多数あることが一目で分かる．以下図6(c)，(d)，(e) は閾値を大きくした場合に描画される画面である．各場合のフレームレートはそれぞれ，50.0，12.2，4.76 であった．図6(e) は省略を行わない場合と同程度の品質の画像が得られるが，フレームレートは約2倍になっている．この理由は，シーングラフ上において着目点から遠い物体，たとえば図4の左端のテーブルや椅子，右端のプロジェクタ等が描画されないからである．

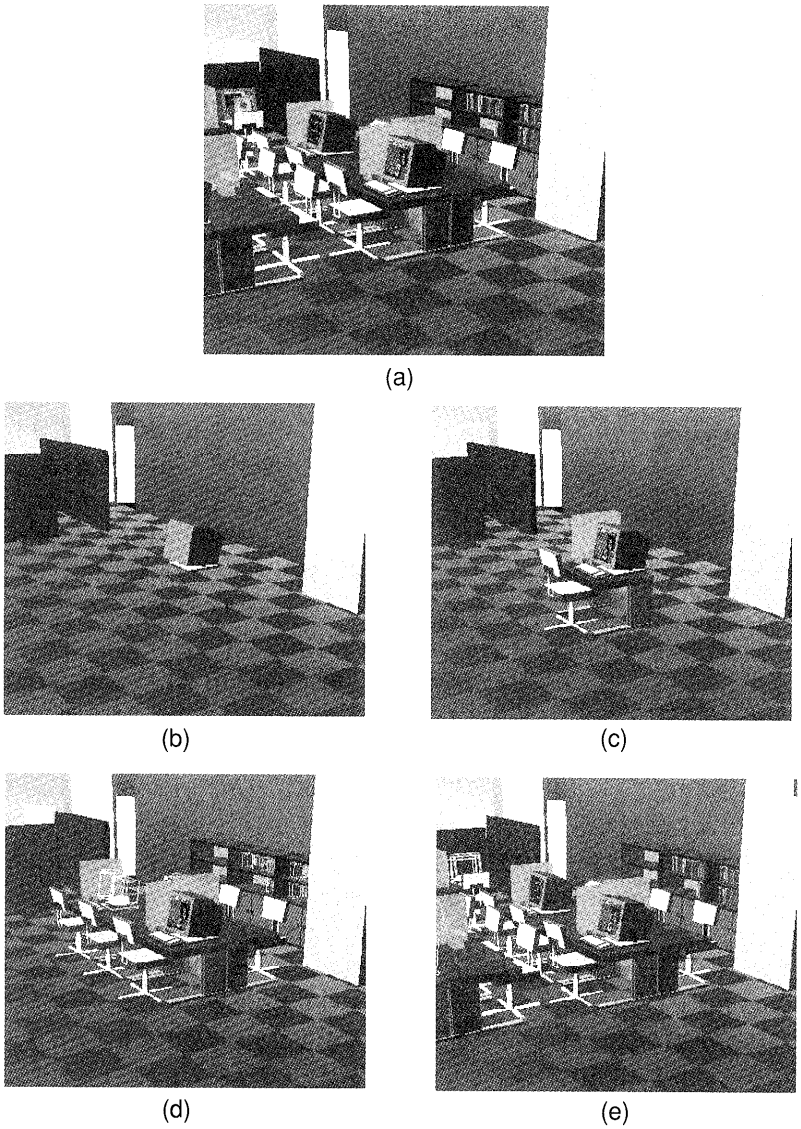


図6 3次元仮想空間ウォークスルーのデモ画面．(a) 全オブジェクトを描画した場合（2.37 frame/sec），(b) 閾値を小さくとった場合（71.43 frame/sec），(c) 閾値を少し増加した場合（50.0 frame/sec），(d) 閾値を中程度にした場合（12.20 frame/sec），(e) 閾値を大きくとった場合（4.76 frame/sec）

Fig. 6 A virtual walkthrough demonstration. (a) all objects are rendered (2.37 frame/sec), (b) rendered with smaller threshold (71.43 frame/sec), (c) rendered with medium threshold (50.0 frame/sec), (d) rendered with large threshold (12.2 frame/sec), (e) rendered with larger threshold (4.76 frame/sec).

5. 階層構造に対する簡略化を利用した応用的手法

これまで対話性の確保について述べてきたが，本章では視点移動時における不自然なオブジェクトの消え方を少しでも解消するための方法について述べる．ここで述べる方法は本手法の特徴である，階層構造に対する簡略化を利用したものである．

5.1 Simple LOD

我々の目的は，モデル作成のコストを増やすことなくシステムの対話性を得ることにあった．実際，本手法では複数のモデルを定義することなしに対話性を得

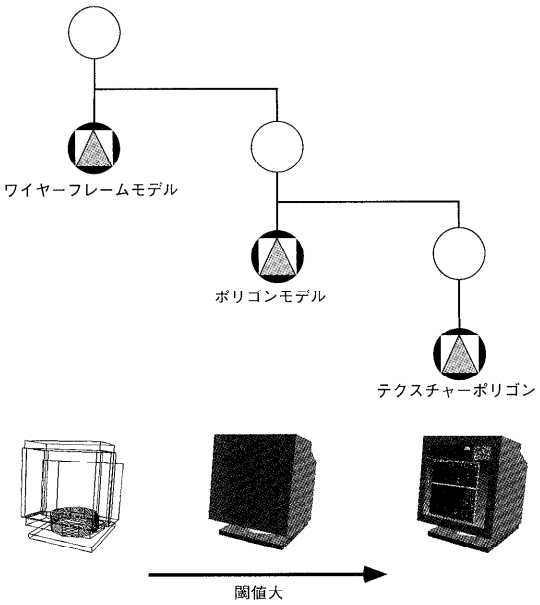


図7 Simple Levels of Detail
Fig. 7 Simple Levels of Detail.

ることができた．しかし，シーングラフに基づくフラクタル値の伝播を利用することによって簡単なLOD機能を実現することもできる．

たとえば，**図7**に示すように，ソリッドモデルで作成されたディスプレイ，ワイヤーフレームのディスプレイ，テクスチャマッピングされたモデル，の3つを用意する．このとき，ワイヤーフレームモデルを親とし，その子供としてソリッドモデルを登録する．さらに，ソリッドモデルの子供として，テクスチャマッピングされたモデルを登録する．このようにモデル化すると，閾値が十分に大きいとき，階層構造の下の方まで描画されるので画面にはテクスチャマッピングされたディスプレイが表示される．逆に，閾値が小さいときには，ワイヤーフレームモデルのディスプレイが表示される．

この方法で簡単なLOD機能を実現することができるが，本手法には限界がある．たとえば，先の例について述べると，閾値が十分に大きくて，テクスチャマッピングされたモデルが画面に表示されているとき，画面にはワイヤーフレームモデルも表示されているのである．ただ，ワイヤーフレームモデルの上を覆い隠すようにテクスチャマッピングされたモデルが描画されているだけなので，実際にはその姿が見えないのである．つまり，この例のように簡単化されたオブジェクトとしてワイヤーフレームモデルを定義するのはよいが，さまざまなレベルのオブジェクトを用意してそれらを使い分けることはできない．ただし，こうした点を理解したうえでモデル化を行えば，効果的なLOD機能を実現することも確かである．オブジェクトの表示もしくは非表示のどちらかしか行わない場合，オブジェクト数やモデリングの方法によっては不自然な消え方をすることがある．これに対しLODを上手に使用すると，こうした不自然さを最小化することができる．実際，図6の仮想空間ウォークスルーの例ではこのSimple LODが用いられている（たとえば，図6(d)奥のディスプレイ）．

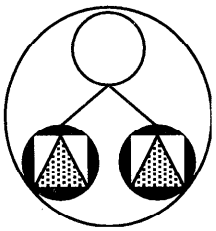


図8 GODのシンボル
Fig. 8 A symbol for GOD.

5.2 Group Of Detail（GOD）

次に，LODの考え方を応用して，Group of Detail（GOD）という手法を今回新しく考案した．LODの基本的な考え方は，1つのオブジェクトに対して複数の詳細度のものを用意し，描画時に適切なレベルのものを表示するというものである．これに対してGODは複数のオブジェクトをまとめて1つのオブジェクトとして簡略表現するものである．GODを用いれば，複数のオブジェクト，たとえば本棚およびその中に入っている本を必要に応じて1つのオブジェクトとして表現することが可能である．または，部屋の中にあるオブジェクト全部を1つにまとめてしまうこともできる．

GODノードはGroupノードとShapeノードの機能を合わせ持つものとして実現した．すなわち，Groupノード同様子供を持つことのできるノードであり，かつ自分自身にShapeを登録することができる．GODノードのシンボルを**図8**に示す．

GODの基本的アルゴリズムは次のとおりである．

- すべての子供のフラクタル値が自分のフラクタル値より大きい場合，自分自身のShapeを描画し，子ノード以下を描画しない．
- 1つでも自分と同じかあるいは小さなフラクタル値を持つ子ノードがある場合，自分自身のShapeの描画は行わず，子ノード以下の描画を行う．

ただし，全体のFractalViewsを制御する閾値はGODノードや子ノードのフラクタル値よりも等しいか大きいとする．

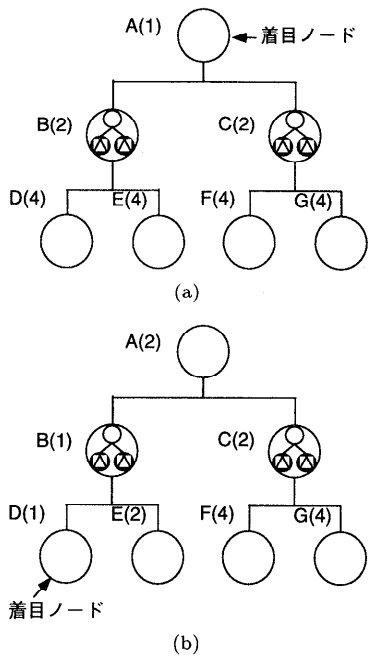


図9 GODのアルゴリズム．(a) 着目点はA，(b) 着目点はD
Fig. 9 An algorithm of GOD. (a) The focus is on A, (b) The focus is on D.

これを**図9**を用いて具体的に説明する．図9において，ノードBとノードCはGODノードであり，閾値は4以上とする．

着目点がAにある場合（図9(a)），GODノードBとその子供，DとEのフラクタル値とを比較すると，$Fv_B < Fv_D$, Fv_E である．このときは，ノードBに定義されているGODのShapeの描画を行い．DとEのShapeについては，閾値が4以上であるにもかかわらず，描画を行わない．同様なことがGODノードCにもいえる．

次に着目点がDにある場合（図9(b)），フラクタル値は $Fv_B = Fv_D$, $Fv_B < Fv_E$ の関係にある．よってGODノードBのShapeは描画されず，ノードDとEのShapeが描画される．GODノードCについては，CのShapeのみ描画され，F，Gは描画されない．

GODに類似した手法は，ConeTree[13]に見られる．ConeTreeでは指定したノード以下の子ノードを表示せず，この部分木を半透明の円錐として表示することができる．我々の手法でもこれと同様の効果が実現できる．たとえば，木の各節点を半透明円錐のShapeを持つGODとして定義すると，着目点をルートにおいた時，閾値が大きければ深い葉ノードまで表示されるが，閾値を小さくすると葉ノードは表示されず，その親ノードの半透明円錐が表示される．

6. 他の応用例

本章では，我々の手法の他の例への応用を示し，その一般性を示す．

6.1 応用例2：CADデータ

最近のCADデータはモデルを構成するポリゴン数が非常に多く，そのすべてを表示しながら対話的に動かすことは難しい．

ビデオゲームや映画等における複雑なモデルの作成を行う際，1つの複雑なオブジェクトを複数の部品（パーツ）に分けて作成している場合が多い．SOFTIMAGE等による代表的3次元モデル作成ツールでは，これら複数のモデルを階層化する手法が一般的にとられている．階層化することで，モデルの管理が容易になったり，モデルにモーションを付ける場合等に便利だからである．

これまでは，こうしたオブジェクトを対話的に動かすためにワイヤーフレームモデルに切り替えて動かすしかなかった．しかし，我々の手法を階層化されたオブジェクトに対してそのまま適用することによって，複雑にレンダリングされた状態においてもオブジェクトを対話的に動かして見ることができる．

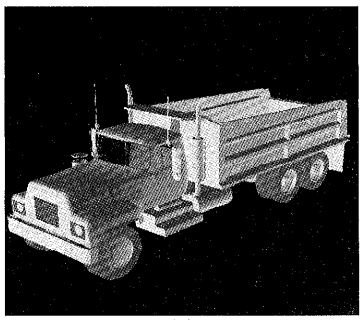
ここでは，**図10**に示すダンプトラックを例に説明する．本データは12287ポリゴンのデータである．着目点はダンプトラックの左のドアに設定してある．図10(a)は視点移動がない状態の画面である．以下図10(b)～(e)になるに従って閾値が大きくなる．フレームレートは図10(a)が10.31 (frame/sec)，以下，図10(b)から(d)はそれぞれ，76.92，71.43，37.04，19.61であった．

6.2 応用例3：分子モデル視覚化

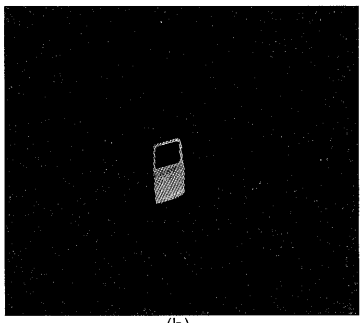
サイエンティフィックビジュアリゼーションの1つに分子モデル視覚化がある．分子モデルを視覚化するさまざまなツール，システムが存在している．VRMLによって分子モデルをWeb上で公開しているサイトも多々ある．ところが一口に分子モデルといっても，球棒モデル，空間充填モデル，化学式モデル，あるいは高分子化合物を抽象化したモデル等，さまざまな表現方法が存在する．

本手法の階層構造に対する簡略化，特にSimple LOD機能を上手に利用することでこうしたさまざまな表現方法，抽象化の度合いを動的に変化させて見せるアプリケーションを作成することができる．

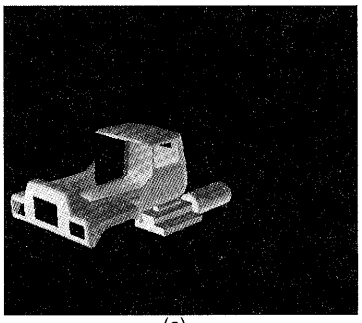
具体的には，**図11**のように空間充填型モデル，棒球型モデル，テキストモデルの3つのモデルを階層化



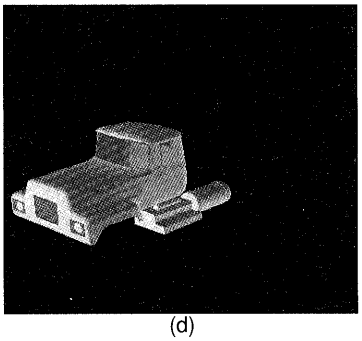
(a)



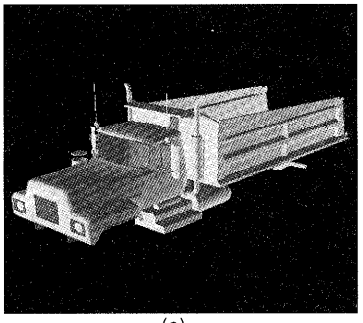
(b)



(c)



(d)



(e)

図 10 ダンプトラックの例. (a) 全部のオブジェクトを描画した場合 (10.31 frame/sec), (b) 閾値を小さくとった場合 (76.92 frame/sec), (c) 閾値を少し増加した場合 (71.43 frame/sec), (d) 閾値を中程度にした場合 (37.04 frame/sec), (e) 閾値を大きくとった場合 (19.61 frame/sec)

Fig. 10 Dump Truck: (a) all objects are rendered (10.31 frame/sec), (b) rendered with smaller threshold (76.92 frame/sec), (c) rendered with medium threshold (71.43 frame/sec), (d) rendered with large threshold (37.04 frame/sec), (e) rendered with larger threshold (19.61 frame/sec).

してシーングラフを作成する. このようにモデル化を行えば, 先に述べた Simple LOD を利用して, さまざまなモデルの切替えを閾値の変更によって簡単に行うことができる.

図 12 は実際にこの方式でグルコース ($C_6H_{12}O_6$) をモデル化したものである. 図 12 (a) は閾値が大きい場合であり, 以下図 12 (b), (c) と閾値を小さくしていった場合に描画される画面である. このように閾値の値に応じて動的にモデルの表現方法を変化させて見ることができる.

今回作成した分子モデルはそれほど大規模なものではない. しかし, これを DNA 等の大規模な分子構造に対して適用することで, 本手法をさらに有効的に利用したアプリケーションを作成できるものと思われる.

7. 考　　察

本手法が着目したシーングラフは現在, VRML をはじめとする多くのグラフィックスシステムにおける

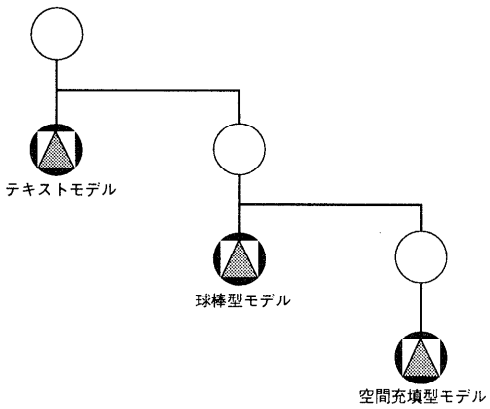


図11 グルコースのシーングラフ
Fig. 11 A scene graph for the grucose.

オブジェクトデータベースとして採用されている．また，本手法は比較的単純なアルゴリズムで実現されているため，本手法の一般性は高いと考えられる．

LOD に代表される従来の手法では，1オブジェクトに対して複数のモデルを定義し，これらの切替えによって描画時間を減少させる方法が一般的であった．これに対して本論文で述べた手法は，描画するオブジェクトの数を制限することによって対話性を確保するものである．

情報視覚化システムで大量の情報を視覚化する場合，個々のオブジェクトは立方体，球といった簡単な形状で表現されることが多い．このように，オブジェクトをそれ以上簡単にしようがない場合，従来のように LOD だけによるアプローチでは限界があり対話性は得られない．しかし，本手法ではこのような場合でも対話性を確保することができる．

また，LOD ではモデルの切替えはモデルを作成するプログラマの側にまかされている．つまり，高性能な計算機のユーザと低性能な計算機のユーザとで，描画の制御は同一であった．これに対し，本手法では閾値の設定の自由はユーザにあり，ユーザは自分の使用する計算機の性能に応じて閾値を設定すればよい．

本手法の欠点の1つは，一部のオブジェクトを非表示にした場合の画像の不自然さである．この問題は，我々の提案した Simple LOD や GOD によって，多少緩和されるが，まだ不自然さは残る．しかし，最初に述べたように，フレームレートと描画するオブジェクト数は一般的に逆比例関係にある．したがって，対話性を第1に考える場合，こうした不自然さはある程度犠牲とならざるをえない．

また，本手法に見られるような一部オブジェクトの非表示による描画の高速化は，従来のゲームソフト等

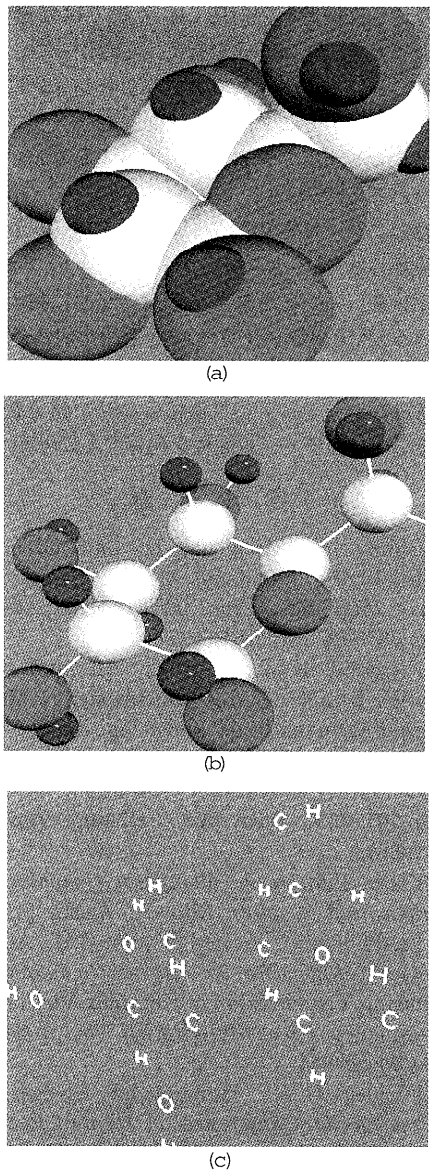


図12 グルコースモデルの実行結果．(a) 閾値大，(b) 閾値中，(c) 閾値小
Fig. 12 Molecular Visualization: (a) larger threshold; (b) medium threshold; (c) smaller threshold.

にも見ることができる．ただし，これらはその場その場でアドホックに解決されていたのに対し，本手法はシーングラフとして表現されたモデルに対し統一的に実現される．

8. おわりに

シーングラフに FractalViews を適用することによって描画シーンの自動簡略化を行う手法を提案した．

今後の展開を以下に述べる．

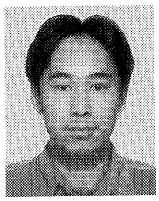
一般のグラフ構造への対応 本手法では階層構造によってシーンの作成を行う．したがって，階層構造にしにくいデータやアプリケーションには適用しにくい．そこで，本手法を一般のグラフ構造に対しても適用できるように拡張することが望ましいと考えている．

VRML ファイルへの対応 現在，VRML が一般に普及しはじめ，VRML フォーマットのファイルがインターネットを通じたデータ交換に利用されている．VRML ブラウザに本手法を組み込むことができれば，大きな 3 次元データであっても利用者のハードウェアの性能に合わせた閾値を設定することによって，対話性を確保することができる．

参 考 文 献

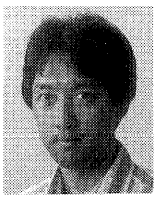
1) Cohen, J., Manocha, D. and Olano, M.: Simplifying Polygonal Models Using Successive Mappings, *Proc. IEEE Visualization '97*, pp.395–402 and 564 (1997).
2) Fairchild, K.M., Poltrock, S.E. and Furnas, G.W.: SemNet: Three-dimensional graphic representation of large knowledge bases, *Cognitive Science And Its Applications For Human-Computer Interaction*, Guindon, R. (Ed.), pp.201–233, Lawrence Erlbaum Associates (1988).
3) Foley, J.D., et al.: *Computer Graphics: Principles and Practice*, Addison-Wesley (1990).
4) Funkhouser, T.A. and Sequin, C.H.: Adaptive Display Algorithm for Interactive Frame Rates During Visualization of Complex Virtual Environments, *Proc. SIGGRAPH '93*, pp.247–254, ACM (1993).
5) Furnas, G.W.: Generalized fisyeye views, *Proc. ACM Conference on Human Factors in Computing Systems* (*CHI '86*), pp.16–23, ACM (1986).
6) 井上正行，小池英樹：Distortion 技術を用いたシーンの自動簡略化手法，日本ソフトウエア科学会第 14 回大会論文，pp.81–84 (1997).
7) 井上正行，小池英樹：フラクタルを用いたシーンの自動簡略化手，グラフィックスと CAD シンポジウム予稿，pp.67–72 (1997).
8) Koike, H.: Fractal Views: A Fractal-Based Method for Controlling Information Display, *ACM Trans. Information Systems*, Vol.13, No.3, pp.305–323 (1995).
9) Koike, H. and Yosihara, H.: Fractal approaches for visualzing huge hierachies. *Proc. 1993 IEEE/CS Symposium on Visual Languages* (*VL'93*), pp.55–60. IEEE CS Press (1993).
10) Leung, Y.K. and Apperley, M.D.: A review and taxonomy of distortion-oriented presentation techniques, *ACM Trans. Computer Human Interaction*, Vol.1, No.2, pp.126–160 (1994).
11) Pesce, M.: *VRML-Browsing & Building Cyberspace*, New Riders Publising (1995).
12) Renze, K.J. and Oliver, J.H.: Generalized Unstructured Decimation, *IEEE Computer Graphics and Applications*, Vol.16, No.6 (1996).
13) Robertson, G.G., Mackinlay, J.D. and Card, S.K.: Cone Trees: Animated 3D visualizations of hierarchical information, *Proc. ACM Conference on IIuman Factors in Computing Systems* (*CHI '91*), pp.189–194, ACM (1991).
14) Sarkar, M. and Brown, M.H.: Graphical fisheye views of graphs, *Proc. ACM Conference on Human Factors in Computing Systems* (*CHI '92*), pp.83–91, ACM (1992).
15) Strauss, P.S. and Carey, R.: An object-oriented 3D graphics toolkit, *Proc. SIGGRAPH '92*, pp.341–349, ACM (1992).

(平成 10 年 5 月 29 日受付)
(平成 10 年 12 月 7 日採録)



井上 正行（学生会員）

1973 年生．1996 年名古屋工業大学電気情報工学科卒業．1998 年電気通信大学大学院情報システム学研究科修士課程修了．同年セガエンタープライゼス入社．アーケードゲームの開発に従事．ヒューマンインタフェース一般に興味を持つ．



小池 英樹（正会員）

1961 年生．1991 年東京大学大学院工学系研究科情報工学専攻博士課程修了．工学博士．同年電気通信大学電子情報学科助手．1994 年同大学大学院情報システム学研究科助教授．現在に至る．1994～1996 年，1997 年 U.C. Berkeley 客員研究員．情報視覚化の研究に従事．特にフラクタルを用いた表示情報量制御手法の開発，情報検索システムへの応用，人工/拡張現実感システムに興味を持つ．1991 年日本ソフトウェア科学会高橋奨励賞受賞．IEEE/CS，ACM，日本ソフトウェア科学会会員．