

〔非公開〕

TR-C-0146

フラクタルを用いた仮想空間中の  
樹木形状の高速表示技術の研究

桑原 教彰  
Noriaki KUWARA

1 9 9 6 3 . 1 5

A T R 通信システム研究所

フラクタルを用いた仮想空間中の樹木  
形状の高速表示技術の研究

平成8年3月

桑原教彰

# 目次

あらまし

1. 序論 .....	1
1. 1 本研究の背景 .....	1
1. 2 本研究の目的 .....	5
1. 3 本研究の概要 .....	6
1. 4 本論文の概要 .....	9
2. 仮想空間表示における課題と本研究での取り組み .....	11
2. 1 まえがき .....	11
2. 2 本研究の位置付け .....	11
2. 2. 1 立体画像表示技術の現状 .....	12
2. 2. 1. 1 立体視特性 .....	12
2. 2. 1. 2 立体画像表示方式 .....	15
2. 2. 2 3次元画像生成技術の現状 .....	17
2. 2. 2. 1 実写画像を用いた仮想空間表示技術 .....	18
2. 2. 2. 2 リアルタイムCG技術 .....	19
2. 2. 2. 3 仮想空間のモデリングに関する技術 .....	26
2. 3 本研究が解決しようとする課題 .....	28
2. 3. 1 3次元画像生成の観点からの課題 .....	28
2. 3. 2 形状モデリングの観点からの課題 .....	30
2. 3. 3 従来手法での課題に対する本研究のアプローチ .....	32
2. 4 まとめ .....	34
3. 樹木形状の高速表示技術 .....	35
3. 1 まえがき .....	35
3. 2 樹木形状のフラクタルによる表現 .....	35

3. 2. 1	フラクタルについて .....	35
3. 2. 2	フラクタルによる樹木形状のモデリング .....	38
3. 3	高速3次元樹木画像生成方法の概要 .....	40
3. 4	フラクタルを用いた樹木形状の簡略化と簡略化形状の多重化 ....	45
3. 4. 1	フラクタルを用いた形状簡略化による形状データ量の減少 .....	45
3. 4. 2	簡略化形状の多重化による形状演算時間の短縮 .....	47
3. 5	実験結果および考察 .....	51
3. 5. 1	形状粗さに対する簡略化形状の形状データの量 .....	51
3. 5. 2	形状粗さに対する3次元樹木画像生成時間 .....	56
3. 6	まとめ .....	60
4.	自然景観の再構成技術 .....	61
4. 1	まえがき .....	61
4. 2	コラージュ定理と形状再構成に使用したフラクタルモデル .....	61
4. 2. 1	コラージュ定理 .....	61
4. 2. 2	形状再構成に使用したフラクタルモデル .....	65
4. 3	多視点樹木画像を用いたフラクタルモデルの推定方法の概要 ....	68
4. 4	最適傾斜法による縮小写像のパラメタ推定 .....	71
4. 4. 1	最適傾斜法について .....	71
4. 4. 2	最適傾斜法を用いた縮小写像のパラメタの推定方法 .....	72
4. 4. 3	最適傾斜法の解の探索開始点の決定 .....	73
4. 5	3次元樹木形状の再構成方法の評価 .....	76
4. 5. 1	フラクタル形状を入力に用いた評価とその結果 .....	76
4. 5. 2	実写樹木画像を入力とした場合の推定結果 .....	89
4. 6	まとめ .....	93
5.	視覚特性を用いた樹木画像生成方法とその評価 .....	95
5. 1	まえがき .....	95
5. 2	テクスチャマップを用いた簡略化形状の表示方法 .....	95

5. 2. 1	テクスチャマップを用いた簡略化形状の表示 .....	95
5. 2. 2	テクスチャマップを用いた簡略化形状の表示方法の評価 ..	98
5. 2. 3	簡略化形状用テクスチャを用いたレンダリング方法 .....	100
5. 2. 4	簡略化形状用テクスチャを用いた樹木画像生成例 .....	103
5. 3	視覚特性を用いた形状粗さの制御方法 .....	106
5. 3. 1	視点からの距離に応じた形状粗さの制御 .....	107
5. 3. 2	視覚特性を用いた形状粗さの制御 .....	111
5. 4	樹木の多数存在する仮想空間表示への適用 .....	117
5. 4. 1	樹木の多数存在する仮想空間表示評価システムの概要 ..	118
5. 4. 2	評価結果 .....	120
5. 5	まとめ .....	125
6.	結論 .....	127
	参考文献 .....	131
	関連するソフトウェアについて .....	139

## あらまし

本論文は、臨場感通信などの人工現実感の応用において、今後、複雑化、高度化していく仮想空間を3DCGを用いて表示する際に、十分な視覚的臨場感と対話性を実現する手法についての研究をまとめたものである。1～2章では、本研究の背景と目的、位置付けについて述べた。本研究では、複雑化、高度化していく仮想空間の例として、特に樹木の多数存在する仮想空間を対象とし、これを表示するための、本研究でのアプローチについて述べた。3章では、樹木のような複雑な形状に対してフラクタルを用いた形状簡略化が有効であることを示し、これを高速樹木画像生成に応用するための簡略化形状の多重化方法について述べた。4章では、3次元樹木形状のフラクタルモデルを、その正面図と側面図をもとにして自動的に再構成する手法を提案し、この手法により、実写樹木画像からそれによく適合する3次元樹木形状が再構成できること述べた。5章では、簡略化した樹木形状を仮想空間表示に用いる場合に、利用者に簡略化形状であることを意識させない表示を可能にする画像生成方法について述べた。6章では、本研究の総括と今後の展望について述べた。

# 1. 序論

## 1. 1 本研究の背景

近年、様々な分野で人工現実感の本格的な応用が模索されている。人工現実感とは、3次元コンピュータグラフィックス(3DCG)などにより生成された画像を2眼式立体表示することで、利用者が実際に存在している環境以外の仮想的な世界(仮想空間)を、あたかもそれが現実の環境であるかのような感覚をもって体験し、かつその仮想空間で行動することを可能とする技術である[1-5]。例えば、人工現実感を電気通信に応用した臨場感通信[6-9]は、単に相手側の映像や音声を伝達するという従来の考え方ではなく、同じ雰囲気を感じられる面談の場、つまり参加者が共有していると感じられる仮想空間を生成し、その中で人と人とが協調的な作業を行なうという考えに立っている(図1.1)。図1.1では、遠隔地に離れた利用者が、3DCGにより生成された仮想の会議室を通信回線を介して共有している。さらに、仮想の会議室中には、同様に3DCGを用いて遠隔地にいる人物を写し込み、面談会議と同じ感覚で通信会議を行なう。

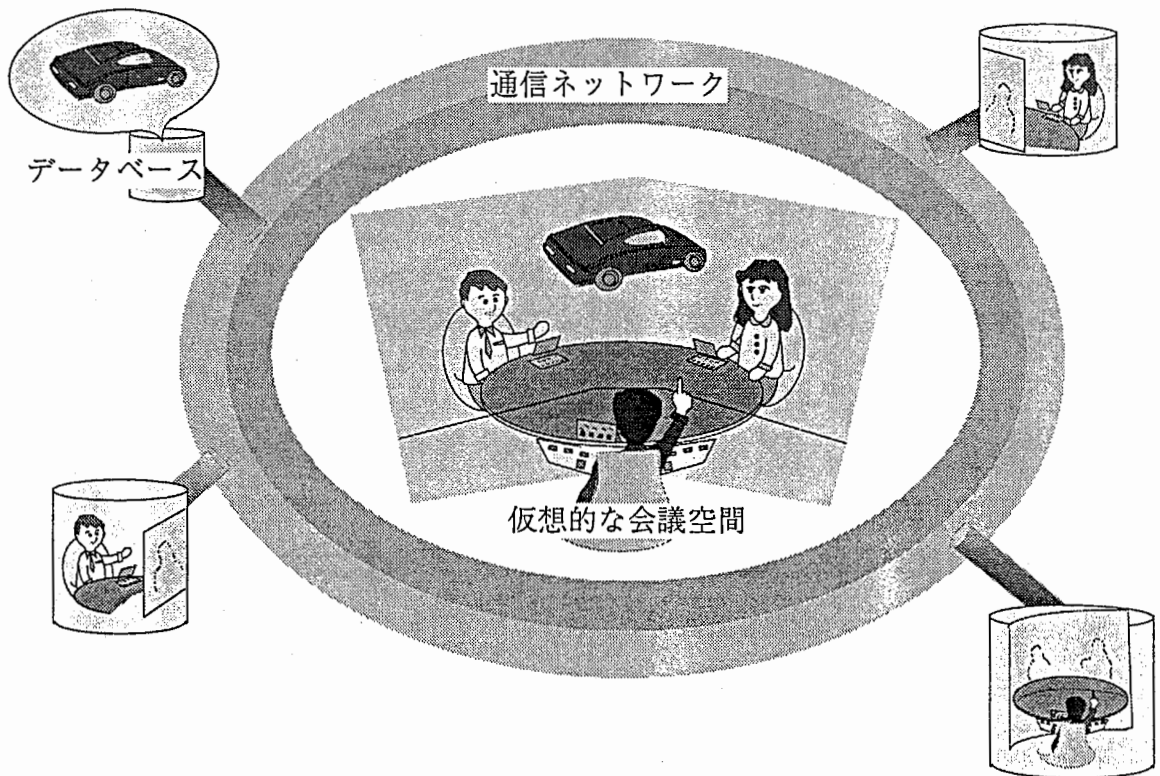


図1.1 臨場感通信会議

このような、仮想空間が利用者にとって現実の環境のように感じられる雰囲気を作り出すためには、仮想空間を利用者に呈示するシステムが、自律性 (Autonomy)、対話性 (Interaction)、臨場感 (Presence) を十分に満足していることが必要であるといわれている [10]。特に、人間は外界の情報の 8 割を視覚系から得ていることから [11]、視覚的臨場感は重要な要素であり、これに関する研究が数多くなされている [5] [12-15]。また、マンマシンインターフェイスの観点から、仮想空間の対話性についての研究が、積極的に行なわれている [16-20]。

視覚的臨場感を生成するためには、(i) 立体画像の表示、(ii) 広視野／高精細なリアリティの高い画像の表示、(iii) 利用者の視点移動に応じた画像を呈示する運動視表示が必要である [5]。例えば図 1. 1 の臨場感通信システムでは、時分割表示される両眼視差画像を液晶シャッター眼鏡を通して見る方式を用いて、立体画像の表示を行ない、また、広視野な仮想空間表示を行なうために、2 面ディスプレイを用いている [9]。一方、リアリティの高い画像生成技術に関して、CG と実写画像の合成による映像作成技法が研究されているが [21-25]、これらは映像製作者サイドの技術であり、運動視表示、すなわち利用者の視点からの画像を呈示することや、利用者と映像の高度な対話性を実現することはできない。このため臨場感通信システムでは、仮想空間中の全てのオブジェクト (仮想物体)、及び人物の 3 次元形状モデルとテクスチャデータを用意しておき、3 DCG を用いた仮想空間生成を行なっている。近年の CG 技術の進歩の結果、CG 画像の写実性が向上し、会議室中の机、椅子と言った単純な形状だけでなく、人物像のような複雑な形状もリアルに表示できるようになった [26-29]。そして、3 DCG を用いた仮想空間表示により、利用者自身の手で仮想物体を掴むなどの高度な対話性を実現している [16-18]。

今後、仮想空間のより高度な視覚的臨場感、対話性を実現するためには、3 DCG を用いた表示技術が、ますます重要な役割を果たしていくと考えられる。例えば、前述の臨場感通信は、B-ISDN の普及に伴い、ネットワークリアリティと呼ばれる高速ネットワーク上に構築された仮想的なコミュニティ空間を、仮想空間として表示することに適用されていくと考えられている (図 1. 2) [30, 31]。また、建築シミュレーション、医療などの分野への人工現実感の応用も検討されており [32-34]、こういった分野で、これまで以上にリアリティの高い、例えば自然景観のように、自然の造形物の多数存在するような仮想空間を表示する技術が要求されている。具体的には図 1. 3 に示すように、樹木、草花といった植物、山、岩石、河川のような地形に関するもの、あるいは雲、雷といった気象に関係するものなどが、仮想空間中に表示されるようになるであろう。しかし一方で、3 DCG を使用する以上、仮想空間中にこのようなオブジェクトが多数存在することにより画像生成に必要な処理時間が増大し、仮想空間の対話性に悪影響を与えることは避けられない [35]。



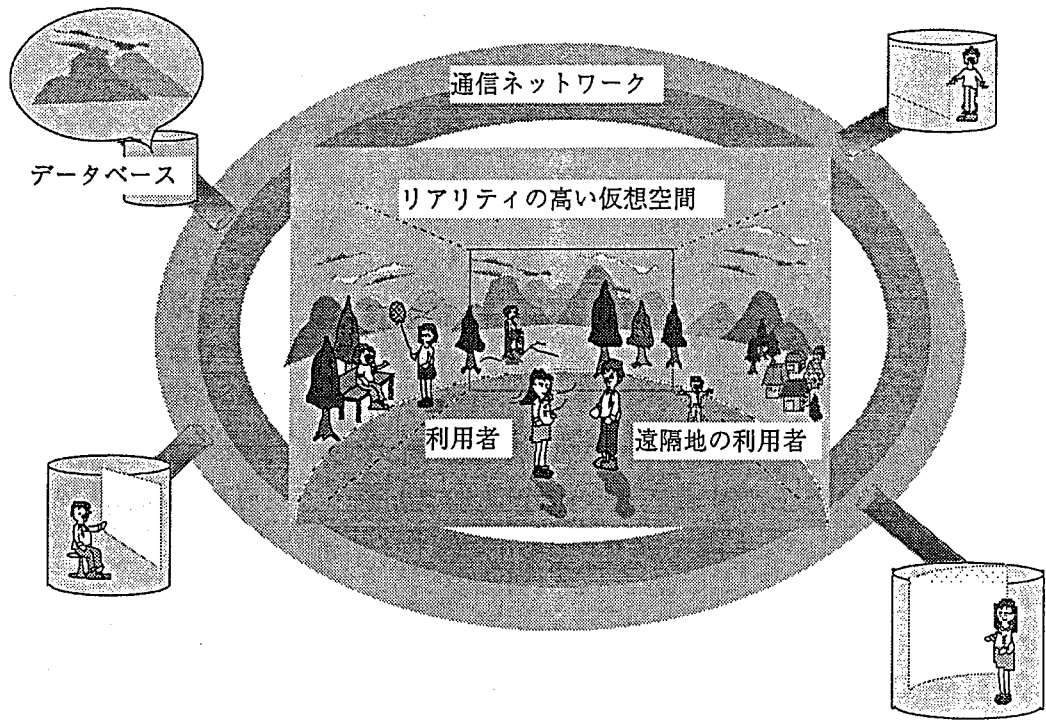


図 1. 2 高度化、複雑化した仮想空間（高臨場感通信）

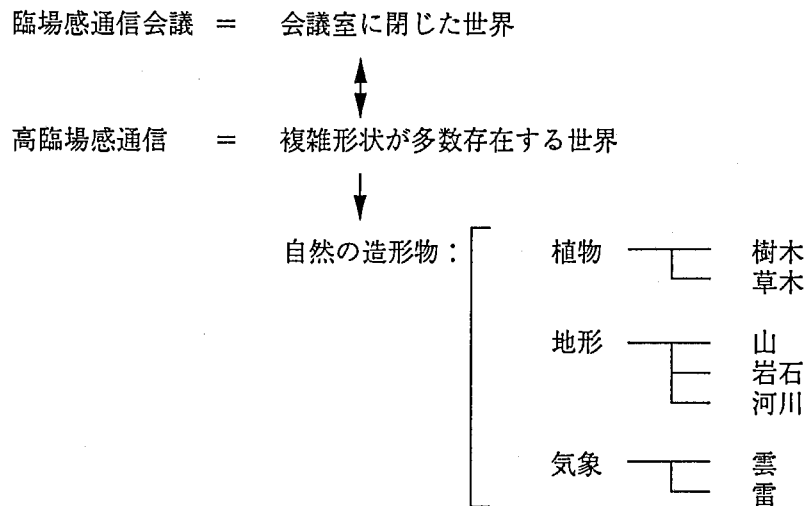


図 1. 3 仮想空間中に表示される自然の造形物

近年のハードウェアの進歩により、CGの描画性能は飛躍的に向上したことは事実であるが、それでも現状では、仮想空間の十分な対話性を実現するために画質を犠牲にしている、あるいは表示する仮想物体を比較的単純なものに制限しているなどが指摘されている [36, 37].

このような背景のもと、本論文「フラクタルを用いた仮想空間中の樹木形状の高速表示技術の研究」は、自然の造形物として樹木を例に取り、図 1. 2 に示すような樹木を数多く含むシーンを対象とした仮想空間表示において、視覚的臨場感と対話性を実現するために必要な、3 DCGに関する要素技術について述べたものである。

## 1. 2 本研究の目的

樹木などの自然の造形物を多く含むシーンを対象とした仮想空間を、3DCGを用いて表示する場合、シーン中の仮想物体の数が非常に多くなり、かつ、それぞれの仮想物体を記述するためのデータ量が膨大となるため、処理時間が増大し、仮想空間の対話性が損なわれる。このため仮想空間表示の際には、リアルタイムCG表示技術が使用される場合がある。

リアルタイムCG表示技術とは、3DCGを用いて表示する物体の形状を簡略化することで、形状データ量を、3DCGで実時間表示可能な程度にまで減少させる手法である [38, 39]。具体的には、例えば描画空間をブロック分割し、視界中の各ブロックに対して、視点から遠ざかるにつれて、形状の簡略化の度合いを大きく設定し、3次元オブジェクトを構成する多角柱に対し、簡略化の度合いとして指定した分解能より小さい凹凸を除去することなどで形状の簡略化を行なうなどといった方法が用いられている。しかし、形状が複雑で多角柱が密に配置されることになる樹木などの形状では、上記の形状簡略化手法では、必ずしも十分にデータ量を削減できない。

建築CADなどの分野においては、樹木の存在する屋外の情景をシミュレーションする必要があることから、樹木の高速な表示方法として、例えば実写樹木画像を形状の単純なポリゴンに対しテクスチャマップする手法が提案されており、実用的な手法としてよく用いられている [40-45]。しかし、この手法では樹木形状の3次元モデルを持たないため、利用者の近くの仮想物体の表示に適用した場合、仮想空間操作による対話的な変形を実現することが難しい。また、風などの外力による揺らぎ [46-48] などを表現することも難しく、こういったことを実現するには、やはり3DCGに頼るほかはない。

そこで本研究では、屋外の情景においてよく現れる樹木を対象とし、これを多く含む仮想空間を、3DCGを用いて十分な視覚的臨場感と対話性を持って表示することを目的とする。具体的には、樹木形状をフラクタル [49-52] を用いてモデル化し、データ量を効率的に削減する形状簡略化により、利用者に画質の劣化を感じさせることなく、シーンの描画に必要なデータ量を実時間表示可能な程度に削減する表示方法を提案する。フラクタルをベースにすることで、本研究で提案する表示方法は、対象とした樹木のみならず、自然界の複雑形状一般を取り扱うように拡張できると考えられる。

### 1. 3 本研究の概要

リアルタイムCGによる仮想空間の表示処理は、図1.4に示すように、実際の形状をモデル化し形状モデルを生成するモデリングの過程と、生成された形状モデルから3DCG用のデータを作成し、それを用いて表示画像の生成を行う画像生成過程の2つの処理に大別できる。本研究の対象である、樹木の多数存在するシーンを表示するには、まずモデリング過程において、実際の樹木に似たリアリティの高い形状モデルを生成することが必要である。そして、その形状モデルを用いて、実時間で表示画像を生成する。

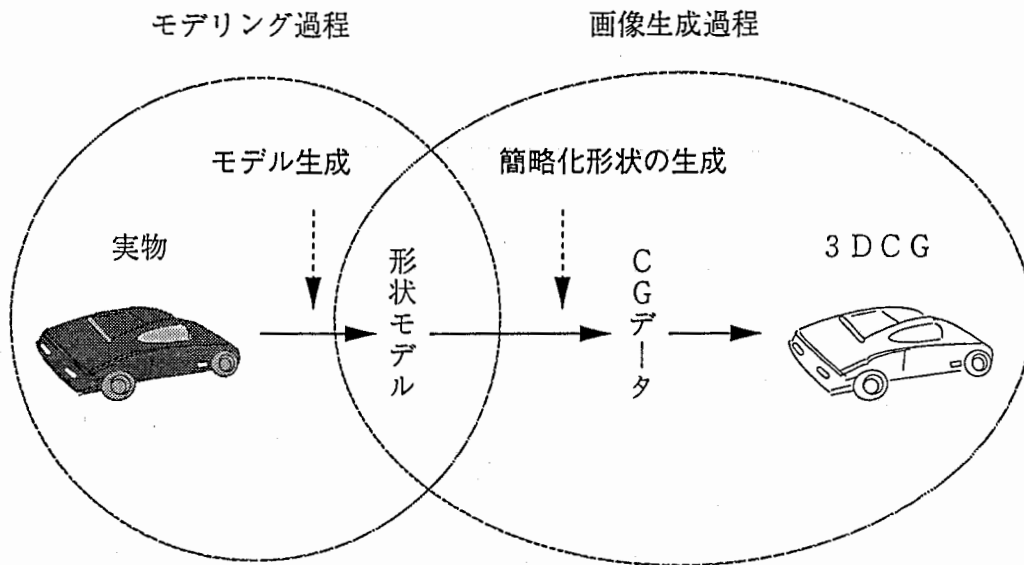


図1.4 リアルタイムCGによる画像生成の過程

こういったことを可能にするための、本研究のアプローチ方法を図1.5に示す。まず、モデリング過程においては、実際の樹木に似たリアリティの高い形状モデルを生成するため、(1) 3次元樹木形状のフラクタルモデルを、その正面図と側面図を用いて自動的に推定する。

つぎに、画像生成過程においては、(2) 推定されたフラクタルモデルを用いて樹木形状の簡略化を行うことで、表示に必要な形状データ量を効率的に削減する。さらに、簡略化形状を多重化することで、フラクタルによる簡略化形状の演算時間を短縮する。また、(3) シーン中の樹木形状を実時間表示可能な程度に簡略化した場合でも、利用者に簡略化形状を意識させない表示を可能にするよう、視覚特性などを用いた3次元画像生成を行う。

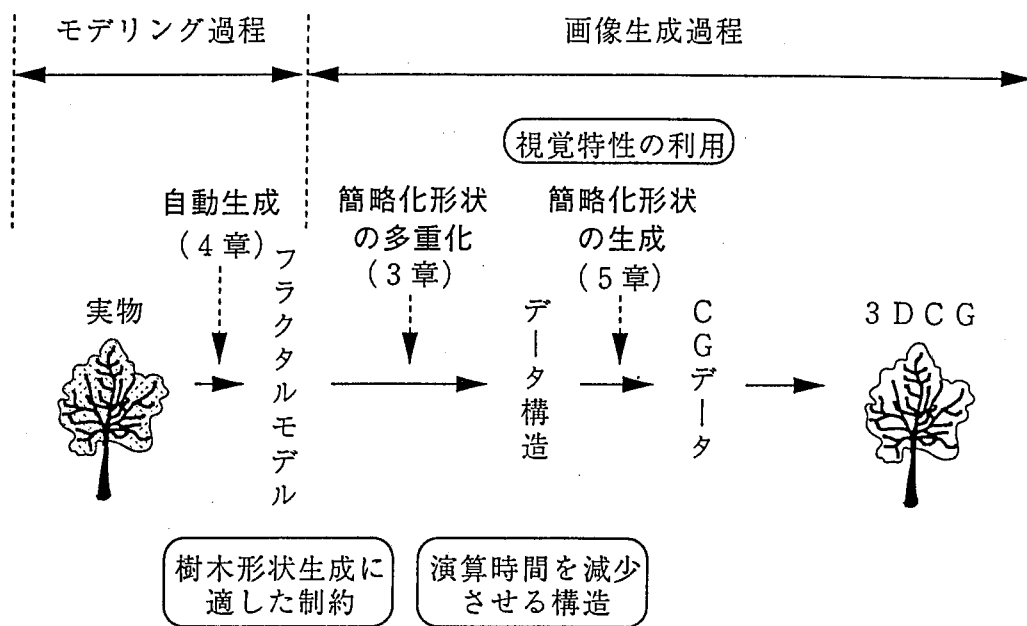


図 1. 5 本研究のアプローチ

(1) 樹木形状のフラクタルモデルの自動推定 [53, 54]

本研究では、樹木の多数含まれるシーンを 3DCG を用いて実時間表示するために、それら形状のフラクタルモデルを用いる。このため、リアリティの高い樹木形状を表示するためには、実際の樹木に似た形状を生成するフラクタルモデルを用意する必要がある。

そこで本研究では、実際の樹木の正面と側面の 2 枚の実写樹木画像から作成したシルエット画像を入力として、それによく適合する形状を生成するよう、フラクタルモデルのパラメタを最適傾斜法 [55] などを用いて推定する。

(2) フラクタルを用いた形状簡略化と簡略化形状の多重化 [56]

形状が複雑な樹木では、樹木形状を構成する多角柱が非常に密に配置されることになるため、視点から遠く離れるにつれて、個々の多角柱を認識できなくなる前に多角柱間の間隔が認識できなくなる。その場合、それら多角柱のグループを 1 つの単純な形状で置き換えることで、データ量を効率よく削減し、かつ視覚的に同等な画質を得ることができると考えられる。そのような効率的な形状のグルーピングに基づく形状の簡略化を、フラクタルを用いることで実現する。従来より、樹木形状を、フラクタルに代表される形状生成規則に基づいてモデリングする研究が行なわれているが

[49-52] [57-64], そのようにモデリングされた樹木形状データが, フラクタルを用いて効率的に簡略化できることについては, これまで十分に検討されていなかった.

また, 上記のフラクタルを用いた樹木形状の簡略化により 3 次元樹木画像を生成する際に, 表示処理に前もって簡略化形状を多重化しておくことにより, フラクタルによる簡略化形状の演算時間を短縮する.

### (3) 視覚特性などを用いた 3 次元樹木画像の生成 [56, 65]

樹木の多数存在する仮想空間を実時間表示可能にするようシーン中の樹木形状を簡略化した場合でも, 利用者に簡略化形状であることを意識させないように, 視覚特性などを用いて 3 次元樹木画像を生成する.

まず, それぞれの簡略化形状に対しては, 樹木画像をテクスチャマップすることで, 形状を大きく簡略化しても, 利用者にそれを意識させない表示が可能であることを利用した, 簡略化形状用のテクスチャデータを用いたレンダリングを行う.

つぎに, 利用者の視点位置などを検出し, それをもとに中心視/周辺視などの視覚特性 [66-68] を利用して, 表示される仮想空間中の各樹木の簡略化形状に与える形状粗さを制御する. リアルタイム CG 技術の分野では, 一般に, 表示面上で一様の表示品質を得るために, 表示面上での形状の最小のサイズが一定となるよう, シーン中のオブジェクトの形状粗さを制御する方法 [38, 39] が検討されているが, 仮想空間表示においては広視野な表示が要求されるため, より一層のデータ量削減を実現する必要があるからである.

## 1. 4 本論文の概要

本論文は、臨場感通信などの人工現実感の応用において、今後、複雑化、高度化していく仮想空間を3DCGを用いて表示する際に、十分な視覚的臨場感と対話性を実現する手法についての研究をまとめたものである。以下に本論文の構成と、各章の概要を述べる。

第2章では、本研究の位置付けに付いて述べ、樹木の多数存在する複雑な仮想空間を3DCGを用いて表示する際に、十分な視覚的臨場感と対話性を実現するための技術的な課題と、それら課題に対する本研究での取り組みについて説明する。

第3章では、樹木形状をフラクタルを用いてモデル化し、その形状を簡略化することで、樹木形状の形状データ量を効率的に削減できることを示す。つぎに、フラクタルによる簡略化形状の演算時間を短縮するための、簡略化形状の多重化方法について述べる。そして実際に、樹木形状のフラクタルモデルを用いて形状の簡略化、簡略化形状の多重化を行うことで、高速な3次元樹木画像生成が可能なことを示す。

第4章では、3次元樹木形状のフラクタルモデルを、その正面図と側面図をもとにして自動的に再構成する技術について述べる。再構成される3次元樹木形状は、第3章で述べる複雑形状の簡略化手法を用いて仮想空間表示を行なうために、フラクタル形状であると仮定し、その正面図と側面図にフラクタル形状を合わせ込むように、フラクタルモデルの縮小写像のパラメタを自動的に調節する。この手法により、実写樹木画像から、それによく適合する3次元樹木形状が再構成できることを示す。

第5章では、樹木の多数存在する仮想空間を実時間表示可能にするようシーン中の樹木形状を簡略化した場合でも、利用者に簡略化形状であることを意識させない表示を可能にする3次元画像生成方法について述べる。

(1) 簡略化形状に対し樹木画像をテクスチャマップすることで、形状を大きく簡略化しても、利用者にそれを意識させない表示が可能であることを示す。また、この考え方による、簡略化形状用のテクスチャデータを用いたレンダリング方法を提案する。

(2) 樹木の多数存在する仮想空間を表示する場合に、利用者の視点位置などを検出し、それをもとに中心視/周辺視などの視覚特性を利用して、表示面上で重要な部分のみを高品質に表示する

ことで、表示に必要な形状データ量が大幅に減少することを示す。

(3) 上記手法により、多数の樹木が風により揺らぐシーンの実時間表示が可能であることを示す。

第6章は結論であり、本研究の内容と成果を概括している。



## 2. 仮想空間表示における課題と本研究での取り組み

### 2. 1 まえがき

本章ではまず、仮想空間表示技術について従来の研究を分類し、本研究の仮想空間表示技術としての位置付けを明らかにする。(2. 2節)。そして、本研究が対象とする樹木の多数存在するシーンを仮想空間表示する際に、従来の研究において課題となる事柄をまとめ、それら課題に対して、本研究がどのような方法で取り組み、解決を計ろうとしているのかについて述べる(2. 3節)。

### 2. 2 本研究の位置付け

あたかも現実の世界のように感じられる仮想空間を生成するためには、利用者の視点移動に応じた立体的な映像を呈示することが重要となる。これまでに、立体画像表示技術に関して、立体画像表示装置、3次元画像生成技術など様々な研究が行われている。現在、立体画像表示装置としては、両眼視差画像を特殊眼鏡を装着することにより、2眼式立体表示する方式が主流であり、また、利用者の視点移動に応じた両眼視差画像を生成するためには、多視点の実写画像に対して画像処理を施し擬似的に視点移動を実現する方式、あるいはリアルタイムCGを用いた方式などが用いられている。本研究は、樹木の多数存在する仮想空間を、3DCGを用いて表示可能にすることを目的としており、図2. 1に示すように、リアルタイムCG技術のなかに位置付けられる。

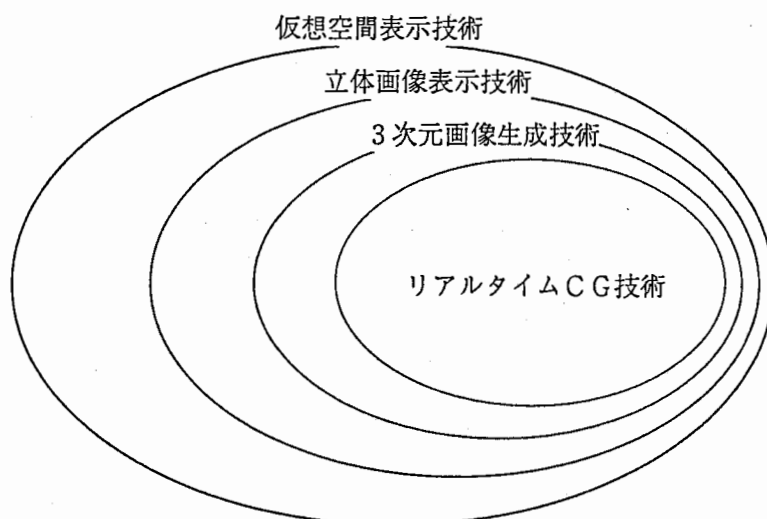


図2. 1 本研究の位置付け

以下では、立体画像表示技術や3次元画像生成技術について、従来、研究されてきた事柄についてまとめる。

## 2. 2. 1 立体画像表示技術の現状

利用者にとって、仮想空間があたかも現実の世界のように感じられるためには、人間が現実の空間を知覚する際の要因に関する知見を元に、それら空間知覚の要因を忠実に再現することが望ましい。そこで本節では、まず人間の空間知覚の要因について述べ、次に、それら要因を利用して立体画像を表示するための、従来の技術についてまとめる。

### 2. 2. 1. 1 立体視特性 [66-68]

表2.1に、人間が3次元空間を認識する際に利用している主な情報と、それが有効に機能する範囲を示す。表2.1に分類するように、3次元空間での位置関係や座標軸方向などの立体情報を検出、知覚する機構には、左右各眼の情報を処理する単眼視機構と、両眼での情報差などを処理する両眼視機構とが存在する。

表2.1 人間の3次元空間知覚要因

	知覚要因	有効距離
単眼視	ピント調節 水晶体の厚さを変える筋肉系の動きと網膜上にできる像のボケ量	2～3m
	像の大きさの変化 網膜上では近くの物体は大きく、遠くの物体は小さい	500m以内
	視野 表示画面が視野に比べて狭いと、立体画像が箱庭のように感じられる	50m以上
	運動視差 観察者または視対象が移動したときに生じる像の動きの相互関係	200m以内
両眼視	両眼輻輳 両眼の内寄せ、外寄せ運動	30m以内
	両眼視差 左右の眼の網膜上にできる像のズレ	100m以内

## (1) 単眼視機構

### (1-1) ピント調節

水晶体の形状を変化させる筋肉系の動きと、網膜上にできる像のボケ状態から判断するため、近距離物体の距離を判別する場合に有効な情報となる。

### (1-2) 像の大きさの変化

等サイズ物体が規則的に分布している状態などで、特に効果的な情報となり、大きさが既知の物体では絶対距離が判定できる。普段の生活では、他の情報からの判断で補正され、かなりの遠距離まで有効である。

### (1-3) 視野

人間がごく自然に見ることのできる範囲は、水平約100度、垂直約80度の広い楕円上の視野になっている。立体映像の表示面が狭いと、表示空間の奥行効果が弱くなり、箱庭的な、臨場感に欠ける映像になる。

### (1-4) 運動視差

観察位置の変化に伴う物体相互の動きや、重なり合いの変化から、前後差の弁別ができ、移動距離や速度を大きくすると遠距離物体感の弁別も可能になる。図2.2に示すように、2点Q1, Q2と観察者Oが与えられ、Oから各点までの観察距離がそれぞれX1, X2であり、しかもOがV<sub>0</sub>の速度で移動しているとき、X1が比較的大きい場合、Q1, Q2がOに与える各速度 $\omega_1$ ,  $\omega_2$ は近似的に、 $\omega_1 = V_0 / X_1$ ,  $\omega_2 = V_0 / X_2$ となり、各速度の差 $\Delta\omega = (X_2 - X_1) \cdot V_0 / (X_1 \cdot X_2)$ を運動視差と呼ぶ。

## (2) 両眼視機構 (図2.3)

### (2-1) 両眼輻輳

両眼の網膜は、写真フィルムのように受光面全体の性能が均一ではなく、中心窩と呼ばれる部分だけが優れた性能を持っている。そのために注視したい物体を中心窩上に結像させるべく眼球を回転させる運動が生じる(輻輳運動)。輻輳運動には、注視動作を始めようとする際に発生する緊張性成分、対象物体にピントを合わせるときに生じる調節性成分、左右眼からの物体情報を単一像として処理できる範囲に持ってくる融像性成分の3成分がある。このうちの調節性成分が、奥行の絶

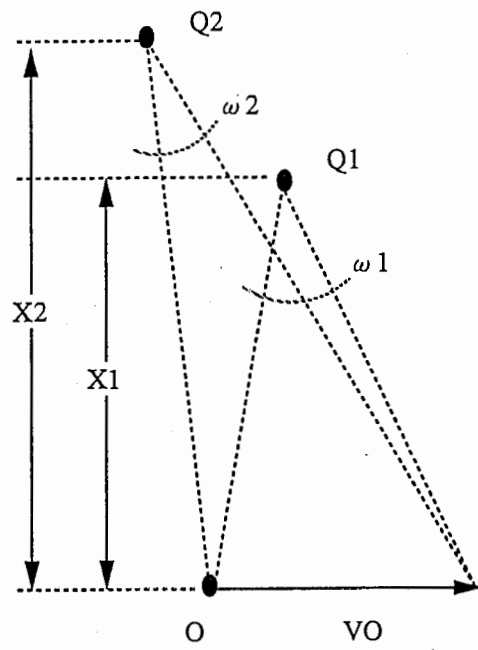


图 2. 2 運動視差

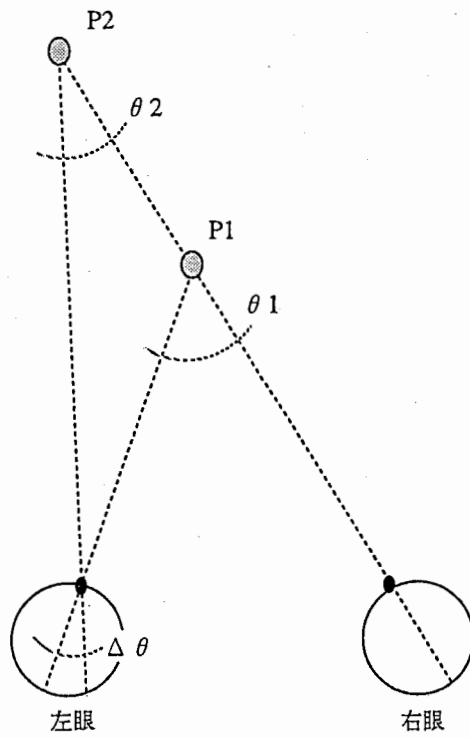


图 2. 3 两眼輻輳と两眼視差

対距離の情報を有効に与える。図 2. 3 に示すように、2 点 P1, P2 が与えられとき、それぞれを注視している場合の両眼の視線のなす角  $\theta_1$ ,  $\theta_2$  を輻輳角と呼ぶ。

## (2-2) 両眼視差

輻輳、ピント調節が働き、両眼に鮮明な物体像が結像されると、その物体を基準として、その前後にある物体は、左右眼の網膜上の異なった位置に結像される。この両眼での像の位置のずれを両眼視差といい、相対距離と前後位置に応じてズレ量とズレ方向が変化する。

図 2. 3 において P1 を注視している場合に、左右眼の網膜上の角度のずれ  $\Delta \theta = \theta_2 - \theta_1$  を、両眼視差と呼ぶ。

単眼視機構と両眼視機構のうち、立体視への寄与度が大きいのは両眼視機構である。ただし、単眼視機構も補助的な働きをするため、自然な立体表示をするためには欠かせない要因である。そこで、観察者にとって自然な立体画像を表示するために、例えば臨場感通信においては、後述する液晶シャッター方式により両眼視差画像を観察することで、両眼輻輳と、両眼視差の要因を満たし、広視野ディスプレイに 3DCG で生成した画像を呈示することで、視野、像の大きさの変化、そして運動視差を実現している。

## 2. 2. 1. 2 立体画像表示方式 [69, 70]

ここではまず、従来の立体画像表示方式を表 2. 2 に分類して示す。表 2. 2 では、立体画像表示技術を、左右の眼に両眼視差画像を与えて立体視する 2 眼式立体表示と、任意の視点から立体像を観察できる 3 次元画像の 2 種類に分類している。

### (1) 2 眼式立体表示

2 眼式立体表示は、液晶シャッター眼鏡などの特殊な眼鏡を装着して、左右の眼それぞれに入る画像を分離することで、両眼視差画像を観察する方式と、特殊な眼鏡無しで両眼視差画像を観察できるレンチキュラスクリーンを用いる方式 [71] に大別できる。特殊な眼鏡を装着する方式としては、赤青眼鏡を用いるアナグリフ式、偏光フィルターを用いる偏光式、液晶シャッター眼鏡で時分割による左右画像の分離を行なう方式が挙げられる。このうち偏光式は、映画館などの多人数、大画面で観察するのに非常に適しており、よく利用されている。また液晶シャッター方式は、家庭用のテレビとの整合がよく、VHD を用いた立体装置が市販されている [72]。レンチキュラー方

表 2. 2 従来の立体画像表示技術

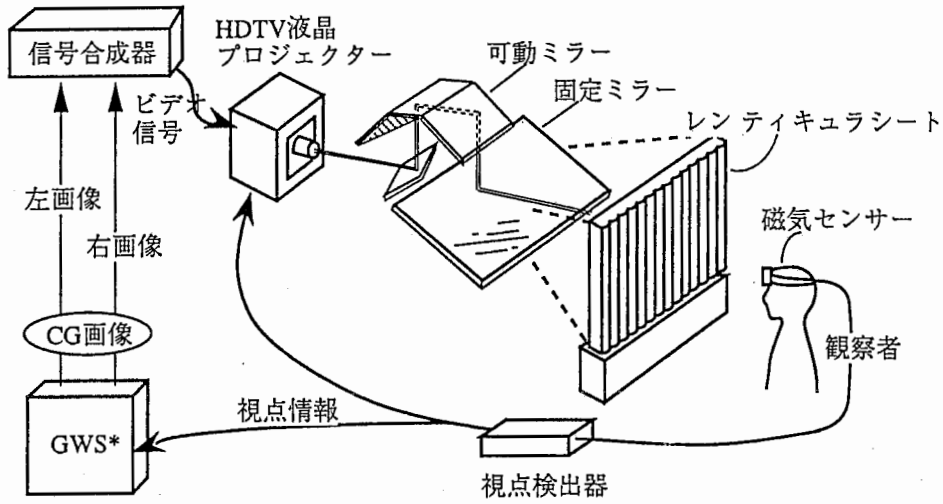
方式		特徴など	
2眼式 立体表示	特殊眼鏡+両眼視差画像	視点移動可能であるが 運動視差は実現不可	
	レンチキュラー+両眼視差画像	視点移動不可	
3次元 画像	ホログラフィ	動画, 大画面化に課題	
	視点追従型 2眼式立体表示	特殊眼鏡+ (3DCGまたは画像処理)	視点移動可能かつ運動 視差も実現(実用期)
		レンチキュラー+ (3DCGまたは画像処理)	大画面化などに課題

式は空間的に左右画像を分離する方式である。

このように、両眼視差画像を用いることで、立体視は可能になるが、視点位置に応じた両眼視差画像が用意されていない場合には、観察位置の移動に伴う映像の変化(運動視差)は感じられない。観察される立体像は、平面像が距離を変えて配置されているような違和感を与えることがあり、舞台の背景に使われる「書き割り」のような見え方であるため、書き割り現象と呼ばれている。これを解決するためには、観察者の位置の変化に応じて適切な両眼視差画像を呈示する必要がある。

## (2) 3次元画像

任意の視点から立体像を観察できる3次元画像方式は、観察者の視点位置を検出して視点に対応した両眼視差画像を生成し、(1)に挙げた方法を用いて視点位置に応じた立体画像を呈示することで運動視差を実現し、疑似的に3次元画像を表示する、視点追従型2眼式立体表示方式と、実像が再生できるホログラフィに分けられる。視点追従型2眼式立体表示方式では、磁気センサーなどにより観察者の位置を検出し、3DCGを用いて両眼視差画像を生成する方式が一般的である(図2.4)[73]。近年では両眼視差画像を、実写画像に画像処理を施して生成する手法についての研究も行なわれている[74]。ホログラフィの動画像技術として、音響光学素子[75]及び液晶パネル[76]を用いたものが報告されているが、大画面の動画像表示を実現するまでには至っていない。



GWS\* グラフィックワークステーション

図 2. 4 視点追従型 2 眼式立体表示方式

以上述べたように、現在は、立体画像表示方式として、特殊眼鏡を装着し左右眼の画像を分離することで両眼視差画像を観察可能にする方法が主流となっている。また、仮想空間表示のように、観察者の視点移動に伴う運動視差を実現する方法については、観察者の視点位置を磁気センサーなどで検出し、それを元に、画像処理、あるいは 3DCG を用いて視点位置に応じた両眼視差画像を生成する視点追従型 2 眼式立体表示方法が一般的であり、技術的にはほぼ成熟したと考えられている。従って今後は、そういった立体画像表示装置を用いて、質の高い 3 次元画像を利用者に提供できるかどうか重要な問題となる。

### 2. 2. 2 3 次元画像生成技術の現状

本節では、3 次元画像を生成する技術について、実写画像を用いる方法、そしてリアルタイム CG による方法について、これまでに研究されている事柄について述べる。

## 2. 2. 2. 1 実写画像を用いた仮想空間表示技術

近年、3DCGにおいて、テクスチャマップ（後述）と呼ばれる、画像データをポリゴンに張り付けることで、単純な形状データを用いながらも十分な質感が得られる技法が、実時間で処理可能になった。これに伴い、実写画像を用いることで、3DCGによる表示処理を軽減したり、景観のモデリング作業を簡易化する手法について研究が行なわれている。例えば、景観中にリアリティの高い樹木画像を高速に描画する実用的な手法として、実写樹木画像を形状の単純なポリゴンに対しテクスチャマップする手法が提案されており、よく用いられている [40-45]。また、景観のモデリング作業を簡易化するために、近景から撮影した景観画像を自動的に領域分割し、いわゆる書き割りを自動生成することで、これを景観の3DCG表示に用いる方法が提案されている [77]。しかし、これら実写画像を用いた手法では、形状の3次元モデルを持たないため、仮想空間中に立体画像として表示する場合、正確な運動視差や両眼視差を表現することは難しい。

一方、上記の3DCGにおける実写画像の利用という流れとは別に、仮想空間の生成に、実際の世界、あるいはカメラからの実写画像を積極的に利用することが検討されている。例えば、遠隔地で機器のメンテナンスを教示するため、シースルー・ヘッドマウントディスプレイ（HMD）を用いて、実際の作業空間に教示用のCG画像を重ね合わせる方法などが研究されている [78-82]。先に述べた臨場感通信システム [9] も、利用者の存在する現実の会議空間と、仮想的な会議空間の融合表示と言える。また、例えば、Virtual Dome [14] の研究においては、カメラから得られた広範囲の実写画像を合成し、HMDを用いて表示することで、リアリティの高い広視野な仮想空間を実現している。さらに、実写画像を用いた仮想空間に高度な対話性を実現する方法として、実写画像を背景表示に用い、利用者周辺の仮想空間は3DCGで表示する、モデル系と非モデル系の融合といった研究が行なわれている [37]。しかし、風による揺らぎ [47, 48] といったような、ダイナミックな自然の景観を背景に使用するには、リアルタイムで動画像を取り込み、利用者の視点位置の変化に応じた画像処理を行なう必要があり、現状では実現することは難しい。また、利用者が仮想空間中をウォークスルーする場合には、広範囲に渡って高度な対話性が要求され、そのような広範囲な領域を3DCGを用いて表示する必要がある。

上記のように、実写画像などを用いる仮想空間表示方法は、いわゆるフォトリアリスティックな仮想空間を利用者に呈示できる点で今後有望と考えられるが、高度に対話的、あるいはダイナミックなシーンを呈示する場合には、現状ではリアルタイムCG技術を利用せざるを得ないといえる。



## 2. 2. 2. 2 リアルタイムCG技術

ここでは、まず3DCG技術一般について説明し、つぎに、3DCG技術において特に、画像生成の実時間性を実現するためのリアルタイムCG技術について述べる。

### (1) 3DCG技術一般 [83]

3DCGとは、コンピュータとグラフィックス周辺装置を用いて、画像や映像を作成する技術のことであり、人間にとって多くの情報を交換している媒体である画像・映像を取り扱うため、多くのコンピュータ応用分野において活用されている。特に、気象、流体解析などによる大規模なシミュレーション解析結果のデータを、分かりやすくグラフィック表示することなどを目的とするサイエンティフィック・ビジュアライゼーション、あるいは人工現実感の分野での画像生成における、重要な要素技術となっている。

3DCGの処理の流れは、図2.5に示すように、モデリング処理とレンダリング処理の2つに大別できる。

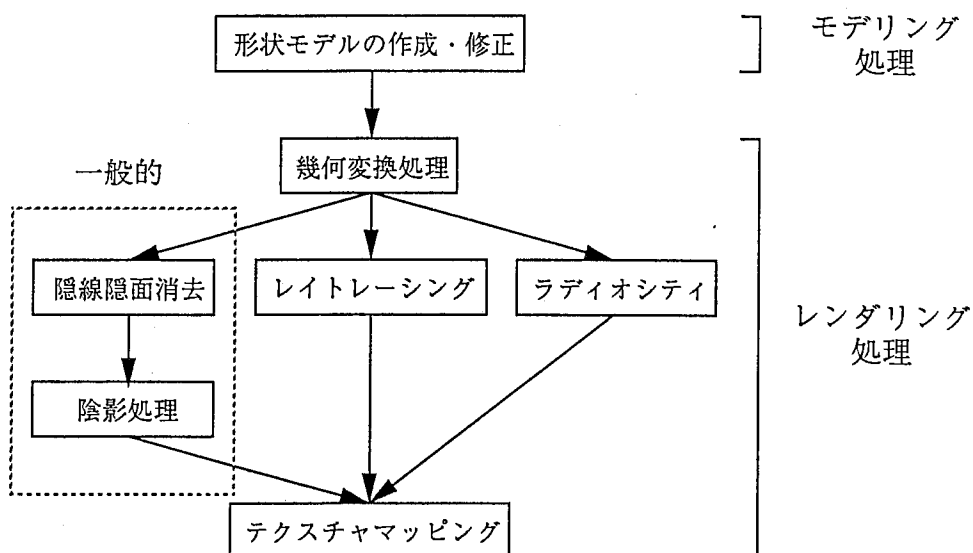


図2.5 3DCGの処理の流れ

### (1-1) モデリング処理

モデリングとは、3次元の形状をコンピュータ上で表現する方法のことであり、ワイヤーフレームモデル、サーフェスモデル、ソリッドモデルなどが挙げられる。CADなどで複雑な3次元形状をデザインするといった、マンマシンインタフェースに関係する作業においては、形状を構成する最小単位を、立方体、球などのプリミティブを合成するというように、ソリッドモデルがよく使用される。一方、リアルタイム画像生成のように高い処理効率が要求されるような場合には、形状を3角パッチのような多角形で表現するサーフェスモデル、あるいは線分で表現するワイヤーフレームモデルがよく使用される。

### (1-2) レンダリング処理

レンダリングとは、モデリングデータを最終的に我々が観測できるように可視化する作業であり、主な処理として、以下に示す(a)～(f)が挙げられる。

#### (a) 幾何変換処理 (図2.6)

モデリングされた3次元物体の形状モデルは、モデル空間という各物体の固有の座標系の上で記述されている。これを、各物体が配置されるワールド座標系に変換し、これを視点位置に対して透視投影変換することで、デバイス座標系に変換する。その結果をスクリーン座標系に対応させる。

#### (b) 隠面・隠線消去

対象とするモデルをディスプレイ上に表示するときには、手前に存在する物体または部分により隠されて、本来は見えない面や線を消去する作業が必要となる。これを隠面・隠線消去といい、3DCGには不可欠な処理である。隠面・隠線消去には様々な手法が存在するが、現在ではZバッファ法が一般的である。

#### (c) 陰影処理

サーフェスモデルで記述された3次元物体に対して、物体周囲の光源や環境条件、そして物体の表面の性質によって、物体表面の明るさや色を決定する方法を、シェーディングあるいは陰影付けと呼ぶ。

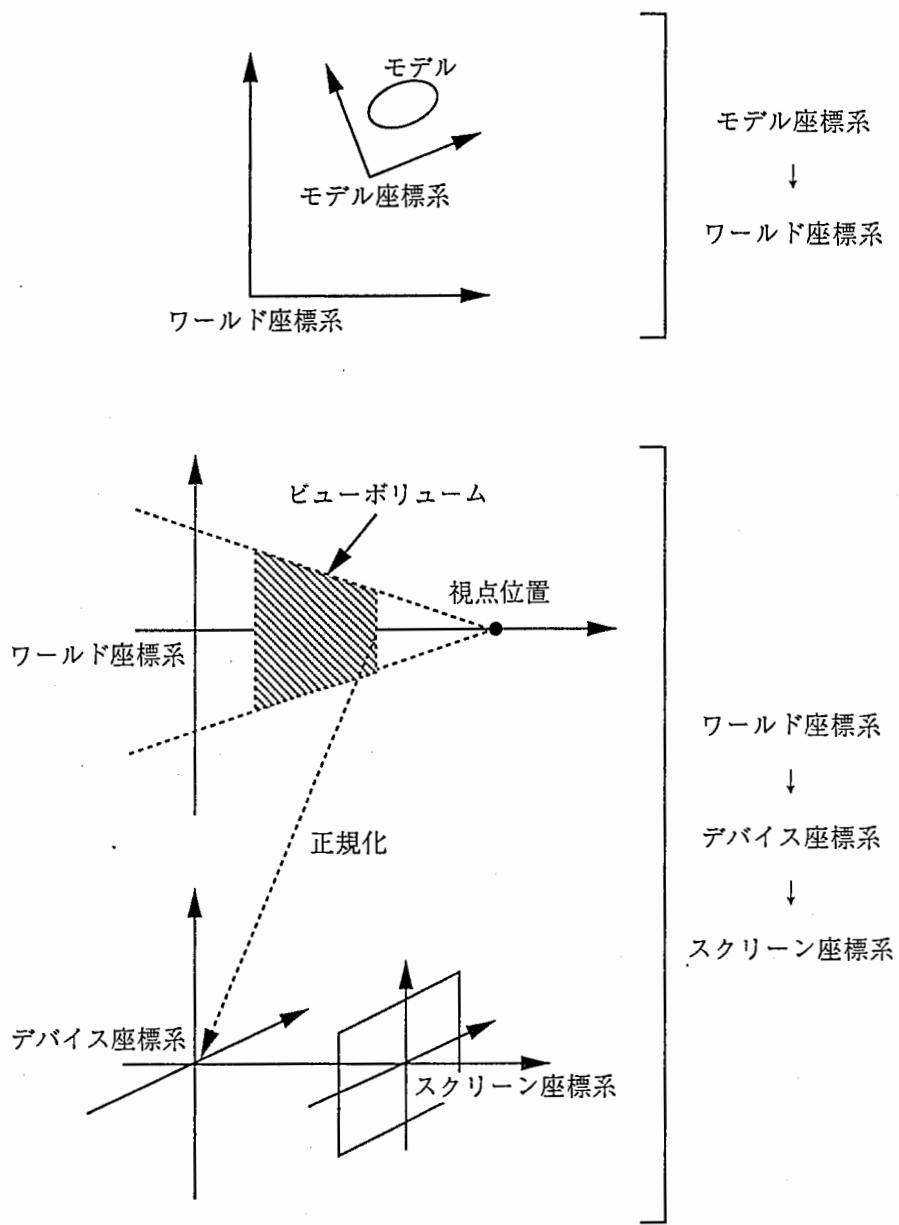


図 2. 6 幾何変換処理

#### (d) テクスチャマッピング

物体表面の複雑な模様や質感を表現するために、例えば物体表面の画像データを、3次元形状モデルの表面に張り付ける方法である。具体的には図2.7に示すように、平面的なテクスチャを、形状のサーフェスに対応させることで処理が行なわれる。また、そのような平面的なテクスチャマッピング以外にも、木目のようなテクスチャを物体の内部にも定義しておいて使用するソリッドテクスチャ [84] と呼ばれる手法が存在する。

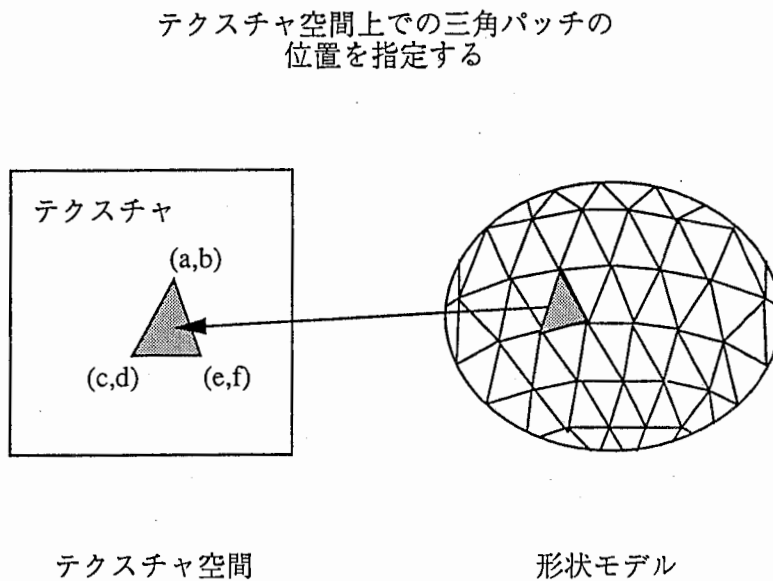


図2.7 テクスチャマップ処理

#### (e) レイトレーシング

レイトレーシングは隠面消去法の一つであるが、光の反射や屈折効果も忠実に再現することができる強力な手法として、現在でも使用されている。レイトレーシングは一般に処理時間がかかるために、多くの高速化技法が提案されている。

#### (f) ラディオシティ

ラディオシティは、環境内に含まれる各種光源と、全ての反射光からの直接の照明量を算出する方法で、極めてリアルな画像のレンダリングが可能となっている。レイトレーシングは、視点の変

化に対して再計算を必要としていたが、ラディオシティは、観察者の視点とは独立になっており、視点移動のアニメーションが容易に作成できることから注目されている。

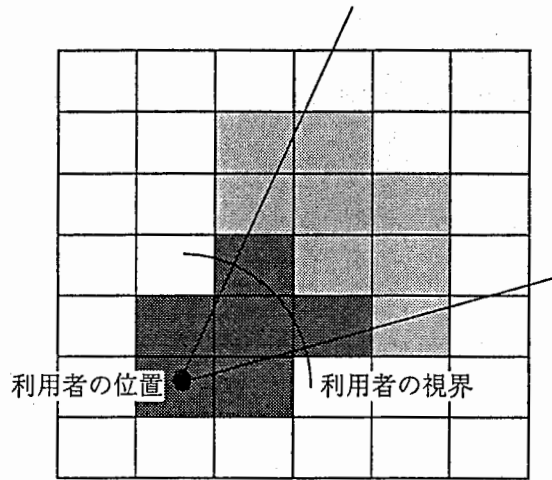
上記のレイトレーシング、あるいはラディオシティは、フォトリアリスティックなCG画像が生成できるため、映像制作現場で一般的に利用されている技術である。しかし、これら技術では実時間での画像生成は不可能であるため、リアルタイムCGの分野でのレンダリング処理は、Zバッファ法による隠線隠面消去、陰影処理、テクスチャマップという流れが一般的である。

## (2) リアルタイムCG技術

従来より、建築CADの分野においては、建物のような人工物に対して、それらの形状を簡略化することでデータ量を削減し、インタラクティブな景観シミュレーションを実現する手法が提案されている [38, 39]。これらは、データを簡略化のレベルを定めるための大域的な手法と、与えられたレベルに応じて3次元オブジェクトの簡略化を行なう局地的な手法を持つ。大域的には、描画空間をブロック分割し、視界中の各ブロックに対し、視点から遠ざかるにつれて、簡略化のレベルを高くする(図2.8)。局地的には、3次元オブジェクトを構成する多角柱に対し、指定する分解能より小さい凹凸を除去することで形状の簡略化を行なう、いわばローパスフィルタによる形状の簡略化手法、あるいは3次元オブジェクトを構成する数種類の基本的な形状に対し、簡略化のレベルに応じて単純化した形状をあらかじめ用意しておき、描画時にそれらの中から適当な形状を選択する(形状の段階的簡略化による多重表現)といった、プリミティブベースの形状簡略化が用いられている(図2.9)。

一方、3DCGによる仮想空間表示において、磁気センサーなどにより検出された視点位置の情報を元に、視覚特性を用いた簡略化レベルの設定を行なう方法についての研究が行なわれている [36]。人間の視覚には、網膜上の視細胞の分布状態により、注視している方向付近は詳細に識別できるが(中心視)、視野の周辺部分では分解能が指数関数的に低下する(周辺視)ことが知られている [85-88]。そこで、階層化適応型3角形パッチ生成法 [89] を使用して、最も精細な形状データから順次精細度を落とした形状データをあらかじめ用意しておき、表示の際には、簡略化レベルに応じた形状データを選択して用いる。階層化適応型3角形パッチとは、指定する分解能より小さい3角パッチを統合して形状を簡略化する方法であり、前述のローパスフィルタによる形状の簡略化手法と基本的な考え方は同じである。

(a) ブロック分割された世界



(b) 描画対象のブロック

詳細 ← 簡略化レベル → 粗

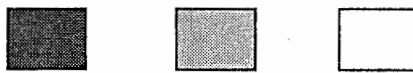
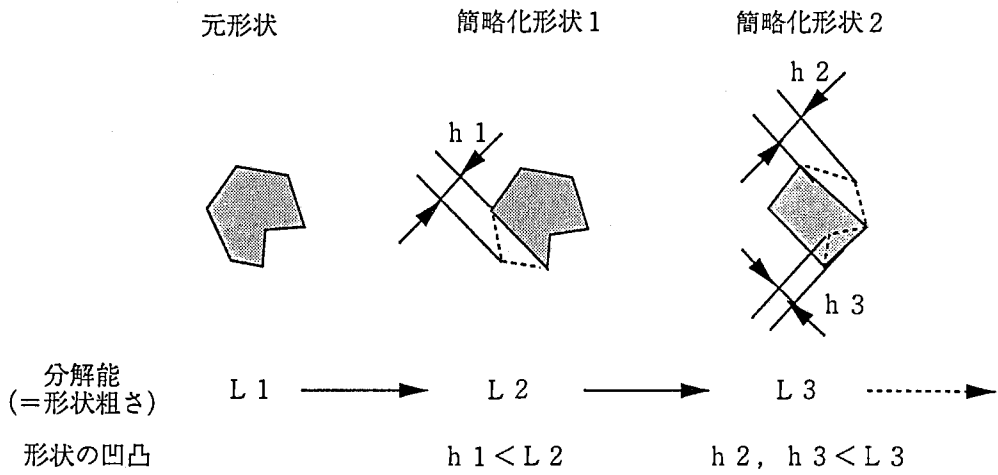


図 2. 8 表示領域のブロック分割管理

(a) ローパスフィルタによる形状簡略化



(b) プリミティブベースの形状簡略化

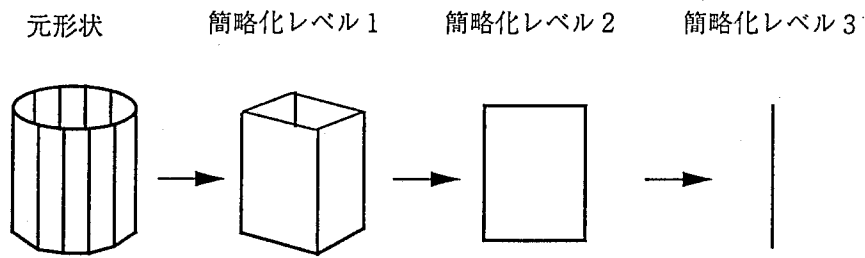


図 2. 9 従来の形状データ簡略化手法の例

## 2. 2. 2. 3 仮想空間のモデリングに関する技術

仮想空間を3DCGを用いて表示するためには、仮想空間中の仮想物体の形状データが必要である。すなわち、これら形状をモデリングしなければならない。例えば、建築CADの分野において、3DCGを用いて景観シミュレーションを行なう場合、新しく建築を行なう建築造形物のモデリングの他に、地形、樹木といった現在の情景をモデル化することが必要である。3DCGに用いる、建築物などの人工物の形状データについては、CADが一般化したことに伴って設計データがオンライン化されたことにより、容易に入手することができる。一方、地形、樹木といった自然物については、実際に測量を行なって地形データを作成する、あるいは、景観の写真をもとにして、地形データ、樹木形状データを手作業で作成することになるが、リアリティの高い形状を得るためには大変な労力を要する。

これまでに、このような自然物の形状を3DCGを用いて表示するための、様々なモデリング技術についての分類が行われている [90-93]。これらによれば、草花、樹木といった植物の形状のモデリングは、以下のように分類される。

### (1) フラクタルなどの形状生成規則に従い樹木形状を生成する方法

これは、例えば樹木のある枝を親にして、その子の枝がどの方向に、どの長さで発生するかを定めた幾つかの規則を繰り返し適用することで、結果として複雑な樹木形状を得る手法である [57-64]。これら手法は、樹木形状を根のほうから順に描く描画手順を与える [90]。

### (2) 実際の植物の成長過程を忠実に再現する方法

これは、(1)の生成規則をベースにして、実際の樹木の観察により、枝の長さ、角度などの統計的なデータを入手して、これに基づいて樹木形状生成を行う方法 [94, 95]、あるいは、日照などの外部の環境とのインタラクションを考慮した方法 [96-98] などが検討されている。

上記手法は、計算機上でのシミュレーションによりリアリティの高い形状を生成することを目的としており、手続き的な形状生成方法として知られている。

一方、実際に存在する樹木の形状を再現するというアプローチも考えられる。例えば、実際の物体のモデリングに必要な形状データを、実写画像から自動的に得る手法についての研究が行なわれており、いわゆる三面図から3次元形状を再構成する手法 [99, 100] や、任意方向からのシルエットを用いた曲面の再構成手法 [101, 102] が提案されている。しかしこれらでは、再構成される形



状を凸と仮定しており、シルエットに現われない凹面に対応できない。この問題を解決するために、連続シルエットを用いた3次元モデルの復元方法 [103] が提案されている。また、鉢植えの樹木形状のシルエット画像から、その3次元形状を再構成する試みも提案されている [104]。しかしこれも、葉の茂った樹木の場合などでは、シルエットに現われない複雑な内部の構造を再現できず、樹木形状データとして不適切な結果を得る。

近年、自然の複雑形状一般を取り扱う手法としてフラクタルがよく用いられており、シルエット法では扱えないような複雑な形状の再構成手法についての研究なども行われている。例えば、フラクタルを用いた表面形状の再構成として、3次元の地表面の再構成が試みられている [105, 106]。また、フラクタル画像符号化 [107] は、各画素  $(X, Y)$  ごとに輝度値  $Z$  が対応する画表面、 $Z = F(X, Y)$  のなす形状を生成する Iterated Function Systems (IFS) を推定するものであり、フラクタルな表面形状の再構成である。また、3次元の樹状形状の再構成の試みとしては、生体の気管支に対して、Computer Tomography による断層画像から気管支の3次元画像をつくり、それをもとに気管支のフラクタル形状を再構成したもの [108] が挙げられる。

## 2. 3 本研究が解決しようとする課題

ここでは、樹木のような複雑形状が多数存在する仮想空間の表示に、従来のリアルタイムCG技術、そしてモデリング技術を適用した場合の問題点を挙げ、本研究がそれら問題点に対してどのように取り組んでいるのかについて述べる。

### 2. 3. 1 3次元画像生成の観点からの課題

これまで述べてきたように、仮想空間表示において要求されていることは、仮想空間のリアリティと対話性である。リアリティを実現する上での重要な要素として、写実性が高いこと、広視野であることが挙げられる。また、対話性を実現する上での重要な要素として、実時間運動視表示、そして仮想空間操作が挙げられる。そこで表2.3に、2.2.2で挙げた3次元画像生成技術を樹木の多数存在する仮想空間表示に適用した場合、これら項目がどの程度満足されるのかについて考察した結果を示す。

表2.3 高速画像生成の観点からの課題

			リアリティ		対話性	
			写実性	広視野	実時間運動視	仮想空間操作
実写画像+画像処理			○	○	△	×
リアルタイムCG	視覚特性利用	プリミティブベース	△	×	×	○
		ローパスフィルタ	×	△	△	○
		本研究	△	○	○	○

カメラからの実写画像を合成することで広視野を実現し、画像処理を用いることで擬似的に運動視差を実現する方式では、写実性、広視野の観点からは問題ないと考えられるが、2.2.2.1で述べたとおり、仮想空間中の仮想物体の変形やダイナミックなシーンの表示を行うことは難しい。

つぎに、視覚特性を用いて仮想空間中の各形状の簡略化レベルを制御し、個々の樹木形状をローパスフィルタによる方式で簡略化する場合を考える。視覚特性として中心視/周辺視を利用することで、注視点を中心とした狭い領域に表示される樹木のみ詳細な形状データを使用し、それ以外の領域に表示される樹木の形状は大きく簡略化するように、仮想空間中の形状簡略化レベルを制御できる。このため、視野を大きくとってもシーン全体の形状データ量は大きく増加することはない。一方、形状データの簡略化は、概念的には図2.10に示すように、指定する形状粗さより小さい形状表面の3角パッチが統合されることで行われる。このため理想的には、形状データは形状粗さの2乗に反比例して減少する。

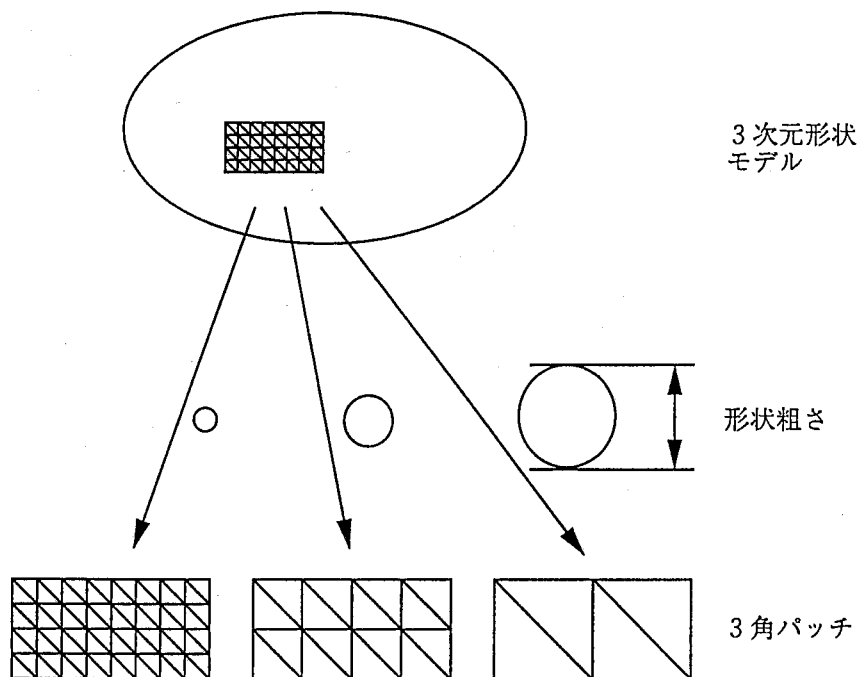


図2.10 階層化適応型3角パッチの例

しかし、樹木のように枝が入り組んでいるような形状は表面積が大きいいため、こういった形状簡略化では、データ量が十分に削減できないことが予想される。さらに、大きい形状粗さを指定してローパスフィルタをかけると、枝の入り組んだ構造が固まりとなって、樹木とは認識し難い形状に簡略化されてしまうため、視覚的に違和感を与えてしまう(図2.11(a))。このため、実時間性を優先して大きな形状粗さを指定した場合、生成画像のリアリティが損なわれると考えられる。

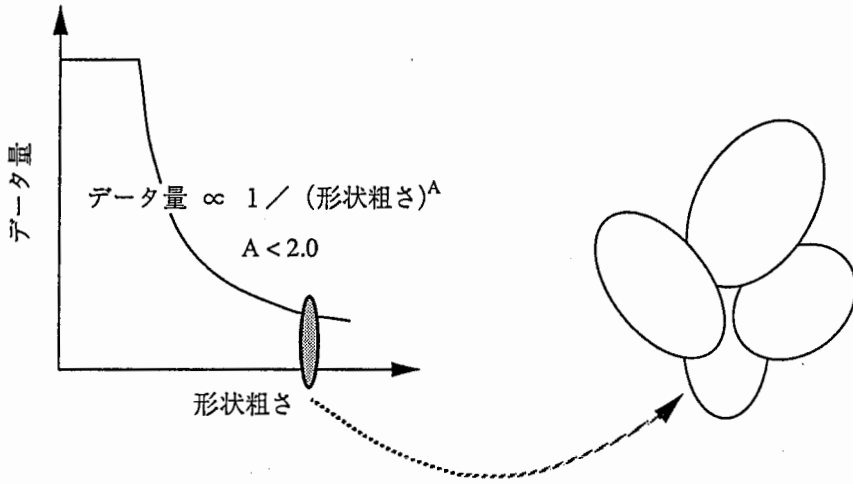
また、プリミティブベースで形状を多重化する方法により樹木形状を簡略化する場合は、簡略化後の形状も樹木らしさを留めることは可能であるが、樹木形状を構成するプリミティブの数が大量であるため、データ量を大幅に削減することはできないと考えられる(図2.11(b))。

### 2.3.2 形状モデリングの観点からの課題

仮想空間に表示する形状のモデリングにおいては、形状のリアリティ、またモデリングに要する作業量、そして生成された形状が実時間表示に適しているかといったことが重要であると考えられる。2.2.2.3で述べた、手続き的な形状生成方法、そして実物の形状再構成方法により、こういったことが満足されているのか、表2.4に示す。

まず、手続き的な形状生成方法における課題について述べる。表示に使用する樹木形状を、実際の樹木のようにリアリティ高くモデリングするには、熟練、そして相当な作業量が要求されるため、フラクタルなどの形状生成規則に基づく方法、そして成長モデルに基づく方法は、それらをできる限り自動的に生成することを目的に研究されてきた。しかしこれら手法により、設計者の意図する形状を生成するためには、様々な成長のパラメタを設定してシミュレーションを行ない、その結果を確認し、更にパラメタを調整するといった、試行錯誤的な操作が必要である。また、個々のパラメタが形状全体の生成にどのように影響を与えるかといったノウハウなども要求されることから、所望の形状を得るためのパラメタの決定を支援する必要がある[91]。さらに、それら手法により生成される樹木形状は非常に精緻であり、そのような形状データを用いて仮想空間表示を行なった場合、現状の計算機の能力では、利用者の視線に追従して実時間で画像を生成することは困難である。例えば、植物成長モデルを景観シミュレーションに応用し、非常に精巧な樹木のモデルを用いて精度の高い景観予測を行う研究が報告されている[109-113]。しかしこれらは、1つの樹木に対して数万ポリゴンのデータを使用し、1つの樹木の描画に数秒を要するといったように、リアルタイムでのインタラクティブな景観表示を目的としたものではない。成長のシミュレーションの精度や形状データの精度を落とすなどの手段で、計算時間やデータ量を削減することが考えられるが、2.3.1で述べたように、樹木のような複雑な形状に対し実時間表示と表示画像の品質を両立さ

(a) ローパスフィルタによる形状簡略化



(b) プリミティブベースの形状簡略化

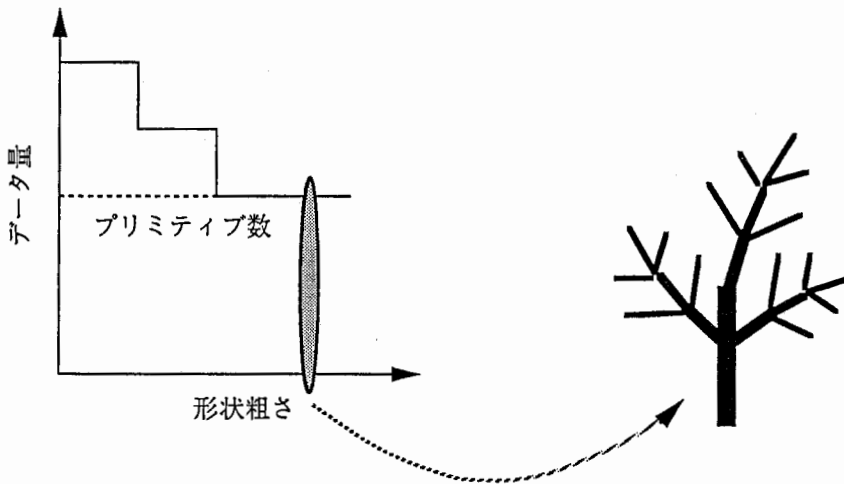


図 2. 1 1 従来の形状簡略化手法による樹木形状の簡略化

表 2. 4 形状モデリングの観点からの課題

		入力	リアリティ	実時間表示	設計の労力
手続き的な 形状生成	成長モデル による生成	成長モデル	○	×	大
	フラクタル モデル	フラクタル モデル	△	×	大
実物の形状 再構成	シルエット法	シルエット	適用不可	適用不可	適用不可
	フラクタル モデル	元形状	適用不可	適用不可	適用不可
	本研究	シルエット	△	○	自動

せることは、従来のリアルタイムCGの手法では困難である。

つぎに、実物の形状を再構成する方法における課題について述べる。まずシルエット画像からの形状再構成方法では、得られる形状は基本的に、連続シルエットを結んだ曲面であり、樹木のように、シルエット内部に微細で入り組んだ構造を持つ形状は、再構成できない。また、従来のフラクタルを用いて複雑形状を再構成する手法は、既知の3次元形状からそれに適合するフラクタルモデルを推定する方法である。

### 2. 3. 3 従来手法での課題に対する本研究のアプローチ

上記のように、樹木のような自然の複雑形状物体を多数含むシーンを仮想空間表示するためには、リアルタイムCGによる大量の複雑形状データを用いた高速画像生成、そして大量の複雑形状データのモデリング、という2つの観点からの課題が存在する。従来の研究では、これらの問題は個別に議論が行われてきたと言える。例えば、リアルタイムCGにおけるローパスフィルタによる形状簡略化手法、あるいはプリミティブベースの形状簡略化手法は、任意の形状データに対して適用可能な汎用的な手法である。しかし、樹木のような複雑形状のモデルの特徴を利用することで、より効果的な形状簡略化が行える可能性がある。また、樹木形状などのモデリングにおいては、リアルな形状の生成が主たる目的であり、実時間表示を考慮した3次元形状モデルにまで言及されることはなかった。しかし、樹木のような自然の複雑形状物体を多数含むシーンを実時間で処理するためには、形状モデリングの過程では実時間処理を考慮した複雑形状モデルを生成し、高速画像生成過程ではその複雑形状モデルの特徴を利用した形状簡略化を行うというような、モデリング過程から

高速画像生成過程までの全体の枠組みの中で問題解決方法を検討することが重要である。

例えば、複雑形状のモデリングにおいてフラクタルが比較的汎用的な手法として用いられているが、フラクタルの特徴を利用して、3 DCGに使用する形状データが効果的に簡略化できること、そして、それをリアルタイムCGに適用することについての検討は行われていない。そこで本研究では、まず、3次元樹木形状をフラクタルを用いてモデル化し、形状を簡略化することで、データ量を効率的に減少させられることを示す。そして、この性質を用いたリアルタイムCG技術について検討する。具体的には、フラクタルを用いて複雑形状をモデリングし、その形状の簡略化形状を生成する。つぎに、簡略化形状を多重化することで、フラクタルによる形状生成の演算時間を短縮し、高速な3次元樹木画像生成を可能にする。そして、広視野な仮想空間表示を可能にするため、形状の簡略化レベルの制御に中心視、周辺視の視覚特性を利用し、また、簡略化形状に樹木画像をテクスチャマップすることにより、大きな形状粗さを指定しても、視覚的な違和感を与えない表示を可能にする。

さらに、上記のフラクタルベースの形状簡略化に用いる、樹木形状のフラクタルモデルの生成方法についても検討を行う。本研究では、3次元樹木形状のフラクタルモデル生成は、実際の樹木形状をもとにして計算機が自動的に行う。実際の樹木の形状は、3次元形状計測器などでその形状データを得ることが不可能であるため、3次元樹木形状のフラクタルモデル生成のための入力として、実写樹木画像を用いる。すなわち、実際の樹木の正面図と側面図から実際の樹木形状のシルエット画像を作り、それによく適合する形状を生成するフラクタルモデルを自動的に推定する。本研究では、これを可能にするための、樹木形状生成に適したフラクタルモデルの制約について検討する。

## 2. 4 まとめ

本章ではまず、仮想空間表示技術における本研究の位置付けについて述べた。すなわち、仮想空間表示において立体画像生成技術が必須であり、その立体画像生成技術のなかでも、仮想空間のリアリティ、対話性を実現するための3次元画像生成技術が重要であることを述べた。そして、3次元画像生成技術について、仮想空間表示の実時間性、そして形状のモデリングという観点から、従来の研究を分類し説明した。

つぎに、従来の3次元画像生成技術の課題として、仮想空間表示における生成画像の写実性、実時間性、そして形状のモデリングにおける形状のリアリティ、形状デザインの作業量などといった項目を挙げた。

そして、本研究では、以下のような方法を提案しそれら課題の解決を計ることを述べた。すなわち、

(1) 3次元樹木形状をフラクタルを用いてモデル化し簡略化することで、データ量を効率的に減少させる。そして、簡略化形状を多重化することで、フラクタルによる形状生成の演算時間を短縮し、高速な3次元樹木画像生成を可能にする。

(2) 広視野な仮想空間表示のために、形状の簡略化レベルの制御に中心視、周辺視の視覚特性を利用する。さらに、簡略化形状に樹木画像をテクスチャマップすることにより、大きな形状粗さを指定しても、視覚的な違和感を与えない表示を可能にする。

(3) 実際の樹木の正面と側面のシルエット画像を用いて、それによく適合する形状を生成するフラクタルモデルを、自動的に推定する。これを可能にするための、樹木形状生成に適したフラクタルモデルの制約について検討する。



### 3. 樹木形状の高速表示技術 [56]

#### 3. 1 まえがき

従来より、樹木のような自然の造形物をフラクタルでモデリングする研究が行なわれているが、フラクタルの性質を利用することで、樹木のような複雑形状が効果的に簡略化できることについては、これまでに十分に検討されていなかった。そこで、本章ではまず、樹木形状をフラクタルを用いてモデル化し、その形状を簡略化することで、樹木形状の形状データ量を効率的に削減できることを示す。つぎに、フラクタルによる簡略化形状の演算時間を短縮するための、簡略化形状の多重化方法を提案する。そして、フラクタルを用いた樹木形状簡略化と簡略化形状の多重化方法の評価として、指定する形状粗さに対する3次元樹木形状データの量、及び画像生成時間について、実際に計算機上で実験した結果を示す。

#### 3. 2 樹木形状のフラクタルによる表現

本節では、まずフラクタルについて説明し、つぎに3次元樹木形状をフラクタルを用いてモデルする方法について述べる。

##### 3. 2. 1 フラクタルについて

フラクタルはマンデルブローが構築した新しい幾何学の対象である [49-52]。フラクタル幾何学は自己相似性を手掛かりにして、非整数次元の世界として知られる複雑な構造を取り扱う方法を与えてくれたため、工学的な分野でも広範な応用を生み出しつつあるのが近年の状況である。特にCGの分野との結び付きは強い。

当初は、フラクタル現象をCGを用いてビジュアライズすることにより、対象を解明しようとする立場から研究が行なわれており、可視化したときのフラクタル画像が非常に美しいことはよく知られている。数学における例としては、複素平面上のフラクタル図形であるマンデルブロー集合が有名である。

一方、自然物や不規則な物体のリアルなCG画像を作成するために、フラクタルを用いるという立場の研究がある。これら研究では、自然物の持つ形状や質感を、コンピュータ上でいかに表現するかという、モデリングあるいはテクスチャ生成技術としてフラクタルが利用されている。例えば、山、川、樹木、草花など、あるいは自然石や鋳造物のモデリング [58] [114]、または、大理石や木目の模様などのテクスチャ生成 [115] などが試みられている。以下では、フラクタルの性質

のうち、フラクタルを用いた複雑形状の簡略化において重要な役割を果たす、自己相似性について述べる。

ある集合を縮小したものが、その集合内の一部分に含まれているとき、この集合は自己相似性を有していると言われる。このような性質を有する集合（自己相似集合）は、縮小写像を介して以下のように定義される。

縮小写像とは、距離  $d$  が定義された空間  $R$  において、空間  $R$  から空間  $R$  への写像  $\omega$  が、 $R$  中の異なる任意の 2 点  $x, y$  に対して、

$$d(\omega(x), \omega(y)) \leq \alpha \cdot d(x, y) \quad (3.1)$$

を満たすことである。ただし  $\alpha$  は  $0 < \alpha < 1$  なる定数である。また式 (3.1) が成立する最小の  $\alpha$  の値を、写像  $\omega$  の縮小率  $\lambda$  という。

さて、空間  $R$  のなかで定義された集合（具体的には図形） $I$  に対して縮小写像を適用すると、 $I$  と相似な図形  $\omega(I)$  に写される。このとき、複数の縮小写像  $\omega_1, \dots, \omega_N$  が存在して、 $\omega_1(I), \dots, \omega_N(I)$  によって元の図形  $I$  が完全に覆いつくされる、すなわち、

$$I = \omega_1(I) \cup \dots \cup \omega_N(I) \quad (3.2)$$

が成立するとき、この集合を縮小写像  $\{\omega_1, \dots, \omega_N\}$  に関する自己相似集合という。この縮小写像の組  $\{\omega_1, \dots, \omega_N\}$  を、フラクタル図形を定義する Iterated Function System (IFS) と呼ぶ。ここで、式 (3.2) を満足する集合は、その大きさを除外すれば、ただ一つ定まる。

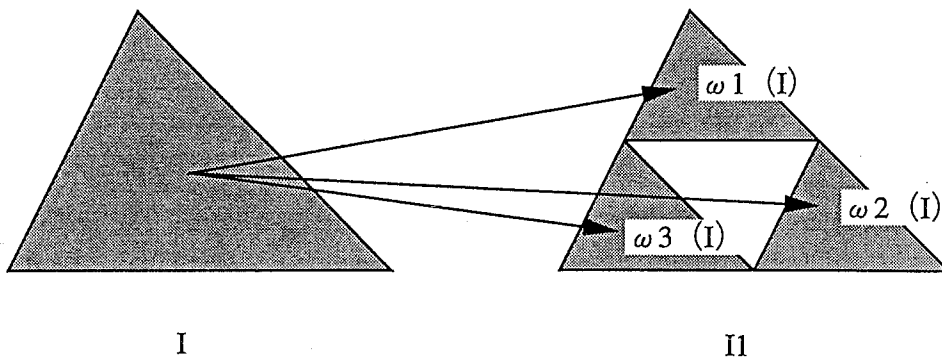
つぎに、縮小写像  $\{\omega_1, \dots, \omega_N\}$  に関する自己相似集合を実際に構成する方法について述べる。まず、空間  $R$  内の任意の誘拐閉集合から出発し、以下の式を繰り返し適用して、集合列  $I_1, \dots, I_M$  を得る。このとき、この集合列の極限集合（アトラクタという） $I_\infty$  は、自己相似集合に一致する。

フラクタル図形の例を図 3.1 (b) に示す。この図形は、図 3.1 (a) に示すように、初期値  $I$  からその 3 つの縮小コピーを生成する操作を、生成された図形に対し再帰的に適用することで得られる。この操作を  $\omega_i$  ( $i=1,2,3$ ) とすると、 $\omega_i$  は 2 次元アフィン変換であり、全ての  $x, y$  に対し式 (3.3) が成り立つ。よって  $\omega_i$  は縮小写像であり、式 (3.2) を無限に繰り返すことで得られる図形はフラクタル図形である。

$$|x - y| < |\alpha(x) - \alpha(y)| \quad (3.3)$$

通常は表示装置の解像度による制約により、適当なところで IFS の適用は打ち切る。また、図 3.1 (b) に示すように、一部を拡大表示するときには、拡大した部分に対し、必要な解像度が得られるまで IFS を適用する。

(a) Generating Fractal



(b) Result of applying  $\omega_1 \sim \omega_3$  to Initial Shape I recursively

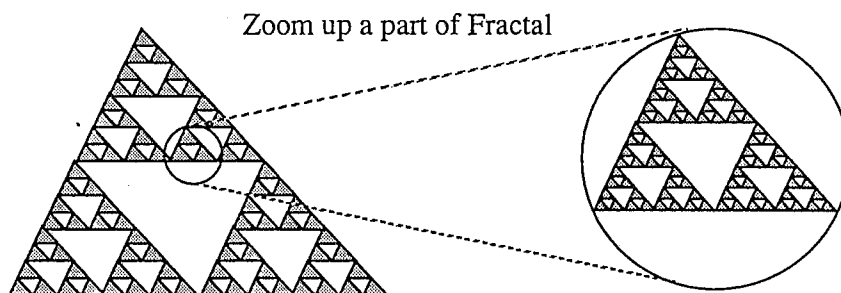


図 3. 1 フラクタル図形の例

### 3. 2. 2 フラクタルによる樹木形状のモデリング

樹木の分岐や葉脈のパターン、つるまき植物の形状などに自己相似性が見られることから、フラクタル図形によってこうした植物の形状をモデル化することが研究されている。このような、フラクタルに従う方法は、単純なアルゴリズムで複雑な形状を生成することができる。例えば、3次元樹木形状をフラクタル図形とみなして形状モデリングを行ない、リアルな樹木画像を3次元CGにより生成することができる。このとき、3次元樹木形状 $T$ を定義する縮小写像 $\omega_1 \sim \omega_n$  ( $n \geq 1$ )は、3次元アフィン変換である。例えば図3.2では、 $\omega_1$ が全体の形状 $T$ を、その一部である $T_1$ に変換することを示している。各 $\omega_i$  ( $1 \leq i \leq n$ )の具体的な表現を式(3.4)に示す。

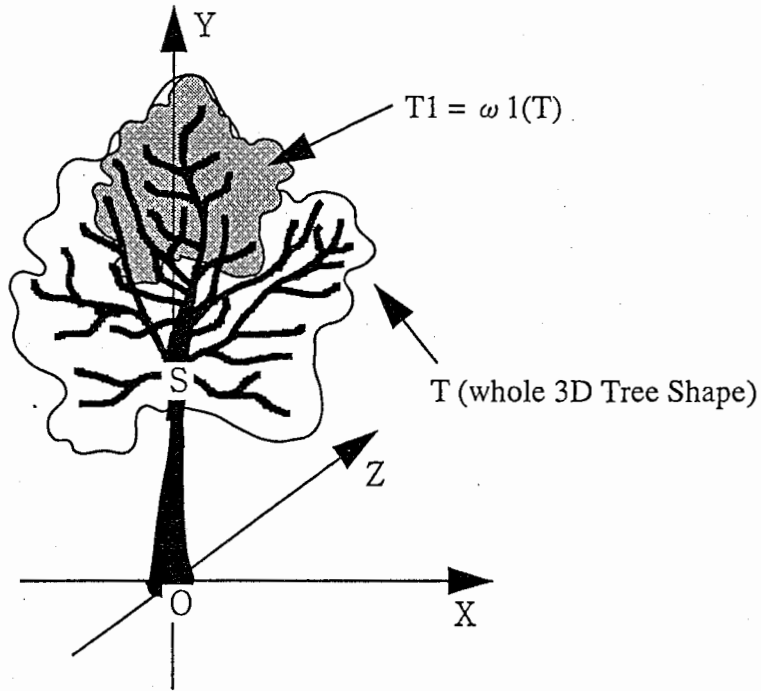


図3.2 3次元樹木形状のフラクタルモデリング

$$\omega = \text{Trans}(0, y_i, 0) \cdot \text{Roty}(\theta_{y_i}) \cdot \text{Rotz}(-\theta_{z_i}) \cdot \text{Scale}(s_i) \quad (3.4)$$

ただし,

Trans (x,y,z)	:	平行移動
Rotz ( $\theta$ )	:	z 軸周りの回転
Roty ( $\theta$ )	:	y 軸周りの回転
Scale (s)	:	拡大縮小

また,  $(y_i, s_i, \theta_{z_i}, \theta_{y_i})$  については,  $y_i$  は枝  $A_i B_i$  が発生する位置,  $s_i$  は幹 OS に対する枝  $A_i B_i$  の長さ比,  $\theta_{z_i}, \theta_{y_i}$  はそれぞれ, 枝  $A_i B_i$  の伸びる方向の z 軸周り, y 軸周りの回転量を表す. このパラメタの組は, 幹 OS の直下の全ての枝について計算される (図 3. 3).

樹木形状 T を生成するには, 幹 OS を初期値として,  $\omega_1 \sim \omega_n$  を再帰的に適用した結果の和集合として得られる (式 3. 5). ただし m および k は適当な自然数.

$$T = \bigcup_{m=1}^n \omega_m(T) \cup OS = \lim_{k \rightarrow \infty} \bigcup_{i=1}^k \bigcup_{(j_1, \dots, j_k) = (1, \dots, 1)}^{(n, \dots, n)} \omega_{j_1} \dots \omega_{j_k}(OS) \cup OS \quad (3.5)$$

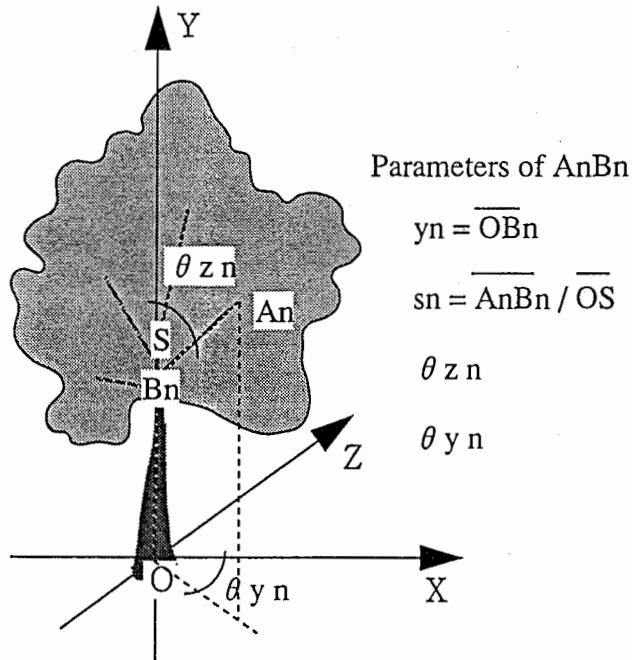


図 3. 3 フラクタルモデルを構成する縮小写像のパラメタ

### 3.3 高速3次元樹木画像生成方法の概要

図3.4に、樹木形状のフラクタルモデル化による、高速3次元樹木画像生成処理の概要を示す。高速3次元樹木画像生成処理は、図3.4に示すように、(1) 樹木の簡略化形状の多重化表現の生成、そして(2) 形状粗さに応じた樹木の簡略化形状の生成の2つの処理からなる。

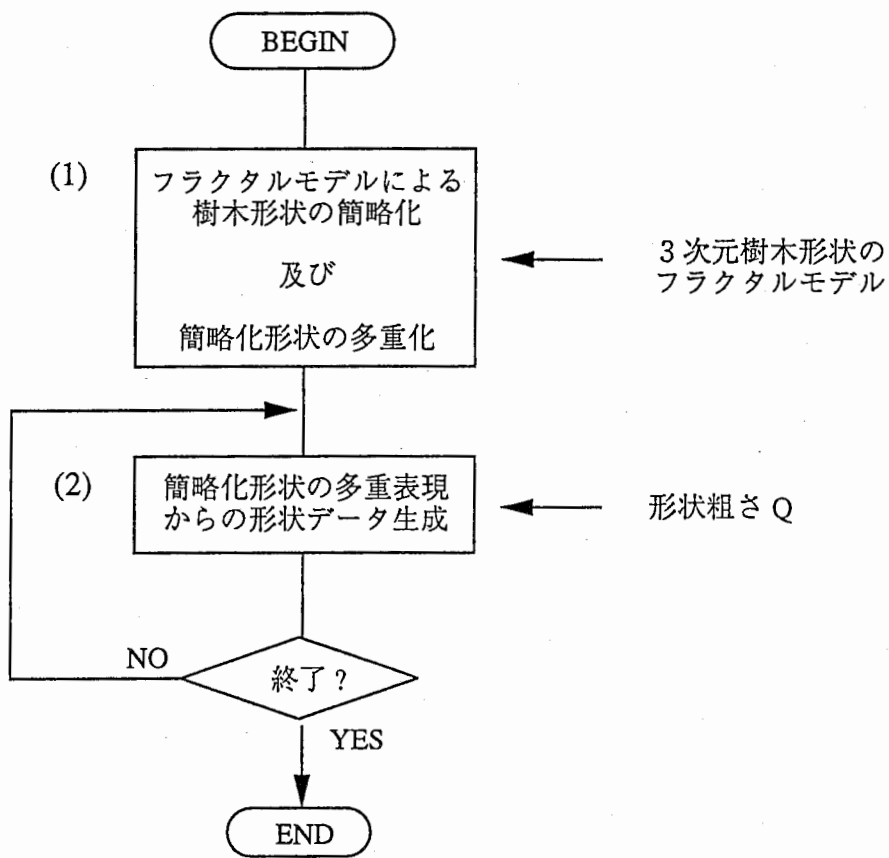


図3.4 フラクタルを用いた樹木形状の簡略化、及び多重化を用いた高速画像生成

### (1) 樹木の簡略化形状の多重化表現の生成

図3.5にフラクタルを用いた樹木形状の簡略化の基本的な考え方を示す。仮想空間中で、樹木が観察者の視点位置の近くにある場合は、その樹木は表示面上に大きく表示される。このとき表示に使用する形状データは、図3.5(a)のような詳細なものが必要である。また、樹木が観察者の視点位置の遠くにある場合は、その樹木は表示面上に小さく表示されるため、図3.5(b)のような粗い形状データを用いればよい。このような詳細度(=形状粗さ)の異なる形状データを、樹木形状をフラクタルを用いてモデル化することで生成する。

上記のように、観察者の視点位置と樹木との距離に応じて、樹木形状の形状粗さを制御する場合、観察者の視点移動の都度、各樹木形状に与えられる形状粗さが変化する。形状粗さが変化する度に、フラクタルモデルにより簡略化形状を演算するのは、画像生成時間の観点から効率が悪いといえる。そこで、樹木の簡略化形状を多重化する。以下で、樹木の簡略化形状の多重化について説明する。

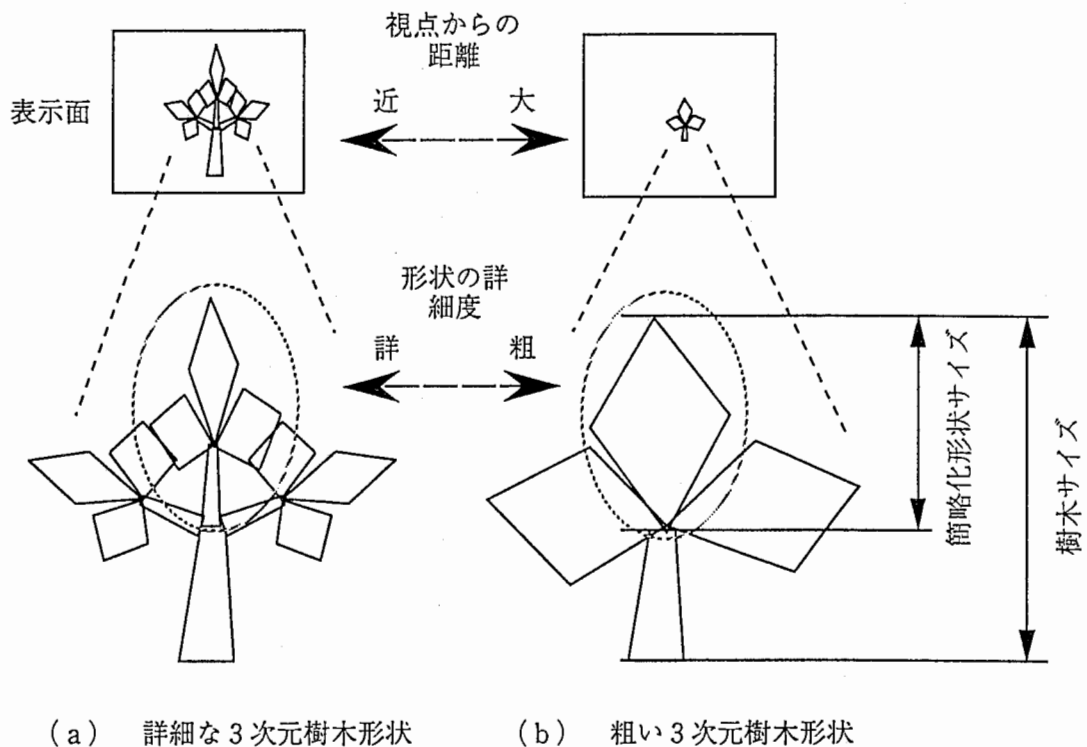


図3.5 フラクタルを用いた樹木形状の簡略化の考え方

まず、フラクタルを用いた樹木形状の簡略化による、形状粗さに応じた簡略化形状のイメージを図3.6に示す。図中の各ノードは、枝データあるいは葉データを表す。各ノード間のリンクは形状データ間の親子関係を表しており、下位の形状は上位の形状から発生することを示す。元の形状は、幹から発生する第1世代の枝、そこから発生する第2世代の枝というように末端まで続き、末端には葉が存在する。また、簡略化レベルNの形状データでは、第2世代の枝以降の形状が、四角のノードで示される代替形状により置き換えられることを示している。そして、簡略化レベルN+1の形状データでは第1世代の枝以降の形状が、さらに、簡略化レベルN+2の形状データで樹木形状全体が、代替形状によって置き換えられることを示している。

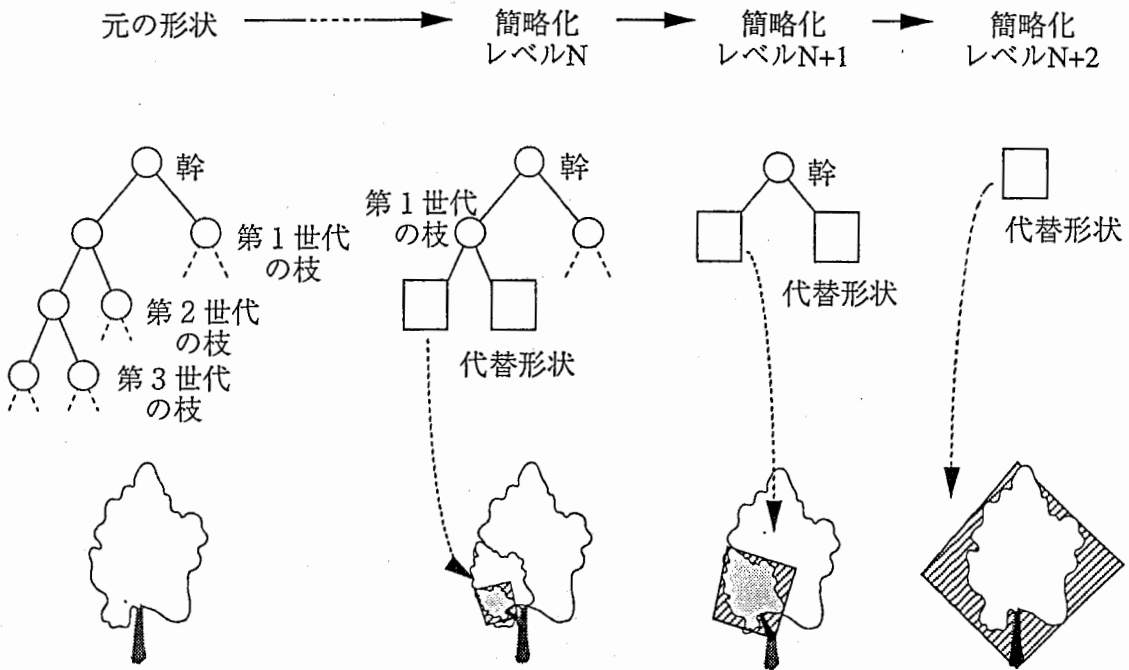


図3.6 形状粗さに応じて簡略化された樹木形状



図3.6に示した、幾つもの形状粗さに応じた簡略化形状を、1つのデータ表現として統合したものを、簡略化形状の多重化表現と呼ぶことにする(図3.7)。図3.7では、ハッチングされた菱形形状が簡略化に用いる形状を表し、菱形内部の形状を置き換える。例えば(1)では樹木全体がa'で置き換えられ、また(2)ではb以下の形状がb'で置き換えられることを表している。このように多重化された簡略化形状を、形状粗さをキーにして1次記憶上に展開しておくことにより、形状粗さが与えられたときに、それを満足する3次元形状を高速に生成することができる。

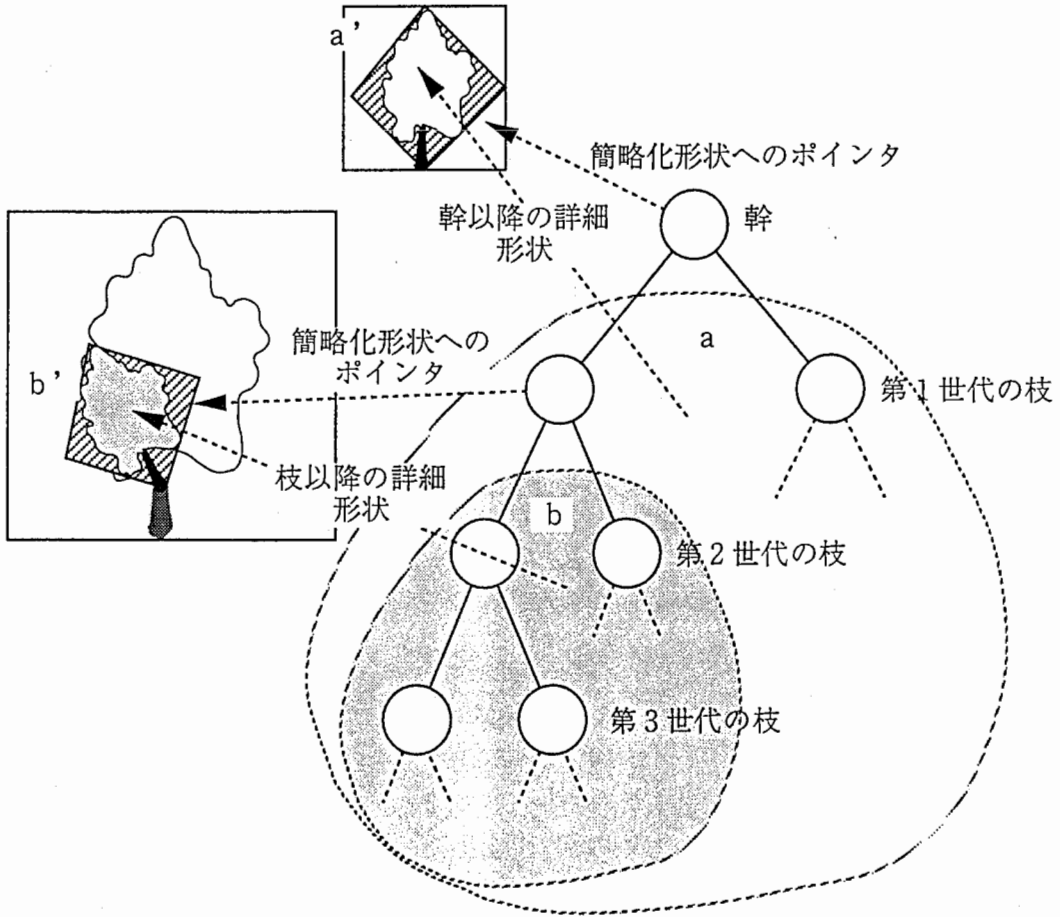


図3.7 簡略化形状の多重化表現

## (2) 形状粗さに応じた樹木の簡略化形状の生成

本処理では、形状粗さが指定されたとき、階層的な樹木形状データ表現から条件に合う3次元形状の構成要素を検索する。検索は樹木形状の元の幹から始まり、その子の枝、さらにそれらの子の枝というように進め、各枝で条件を満足すれば終了する。本処理により、生成される形状データの個数に等しい回数の検索で、形状データが生成できる。

本処理では、形状粗さ $Q$ は式(3.6)に示すように、表示画面上での樹木形状データのうち最小の形状データの表示サイズ $S_{min}$ と、樹木の表示サイズ $S_{tree}$ の比で表すことにする(図3.8)。図3.8では、枝の末端の $S_{min}$ より小さい形状が、菱形のポリゴンで置き換えられていることを示している。

$$Q = S_{min} / S_{tree} \quad (3.6)$$

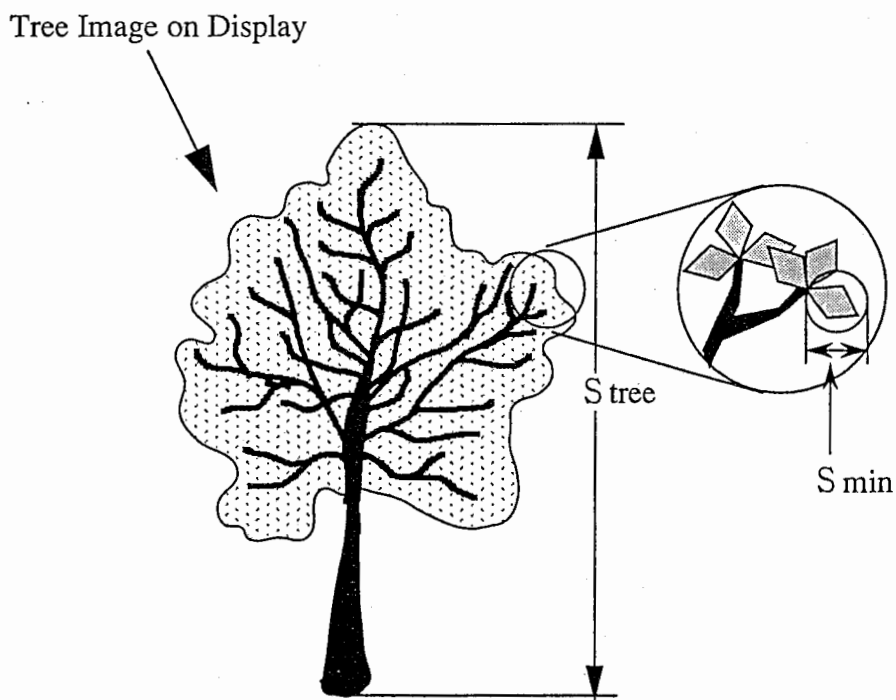


図3.8 形状粗さ ( $S_{tree}$  と  $S_{min}$ )

### 3. 4 フラクタルを用いた樹木形状の簡略化と簡略化形状の多重化

本節では、フラクタルを用いることで、樹木形状が効果的に簡略化できることを明らかにする。つぎに、簡略化形状を演算する時間を短縮し、高速に3次元画像を生成するための、簡略化形状の多重化方法を提案する。

#### 3. 4. 1 フラクタルを用いた形状簡略化による形状データ量の減少

図3. 9は、枝Aから新しい枝A1(図3. 9で黒い枝)が生成される場合を示した図である。A1は式(3. 7)で計算できる。ただし*i*~1は枝ごとに定まる適当な自然数。

$$A_1 = \omega^i(A) = \omega \cdots \omega(OS) \quad (3. 7)$$

枝A1の子孫の枝をA1*m* (*m* ≥ 1) とすると、縮小写像の性質から式(3. 8)が成立する。ただし*p*は枝A1の全ての子孫の枝の個数である。

$$\bigcup_{m=1}^p A_{1m} \subset \omega \cdots \omega(T) \quad (3. 8)$$

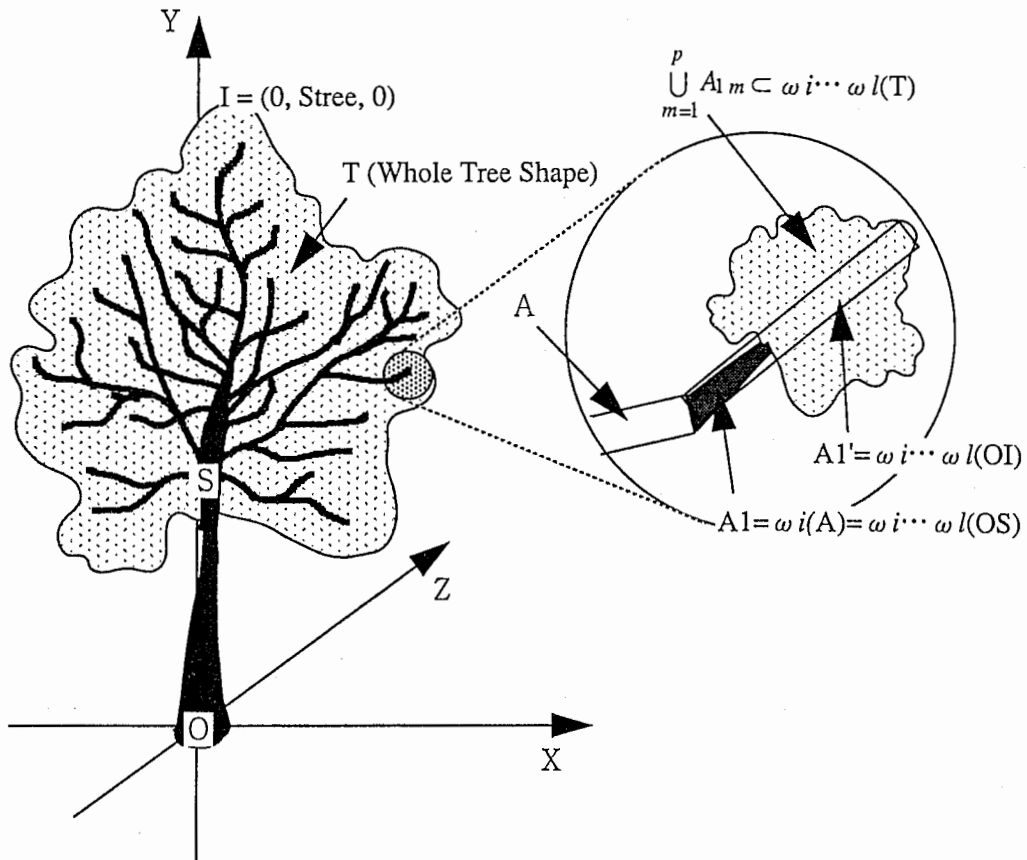


図3. 9 フラクタル樹木形状の枝の生成

ここで、 $\omega_1 \cdots \omega_1(T)$  が表示装置を見ている人が認識できる形状粗さより小さい場合、すなわち式 (3.9) が成立するとき、更に IFS を適用し精細な形状を求める必要はなく、 $\cup A_{1m}$  を  $\omega_1 \cdots \omega_1(T)$  に含まれる単純な形状で置き換えることができる。

$$\forall x, y \in \omega_1 \cdots \omega_1(T) \Rightarrow |x - y| \leq S_{\min}(= Q \cdot S_{tree}) \quad (3.9)$$

以下議論を単純にするため、 $\cup A_{1m}$  の置き換えは枝  $A_{1'}$  ( $= \omega_1 \cdots \omega_1(OI)$ ) (図 3.9) で行われたとし、形状データの量として生成された枝の個数を用いる。ただし、 $I = (0, S_{tree}, 0)$  とする。形状粗さの判定は枝の長さ  $|A_{1'}|$  を使い、式 (3.9) を簡単にした式 (3.9') で行なう。

$$|A_{1'}| \leq S_{\min}(= Q \cdot S_{tree}) \quad (3.9')$$

初期値の幹 OS から発生する枝を第 1 世代の枝とし、第 n 世代の枝から発生する枝を第 (n+1) 世代の枝と呼ぶことにする。また、

- D : 生成された枝の最大の世代数
- N : 生成された形状データの量 = 枝の数
- Q : 指定する形状粗さ
- M : IFS を構成する縮小写像の個数 ( $> 1$ )
- $\lambda$  : IFS を構成する縮小写像の縮小率の平均 ( $< 1$ )

とする。このとき各枝から M 本の枝が発生するので、第 i 世代の枝は  $M^i$  個存在する。よって第 D 世代までのデータ量 N は、

$$N = \sum_{i=0}^D M^i = \frac{M^{D+1} - 1}{M - 1} \quad (3.10)$$

となる。そして、 $M^{D+1} \gg 1$  であるから

$$N \approx \frac{M^{D+1}}{M - 1} \quad (3.11)$$

とすることができる。また、IFS の適用毎に  $A_{1'}$  の長さは平均で  $\lambda$  ( $< 1$ ) 倍され、第 D 世代にはその長さはほぼ  $S_{tree} \cdot \lambda^D$  になる。このとき式 (3.9') を満足し、その枝の長さが  $S_{\min}(= Q \cdot S_{tree})$  となるのであれば、与えられた形状粗さ Q に対して以下が成り立つ。

$$Q = \lambda^D \quad (3.12)$$

よって以下の関係を得る。

$$N \approx \frac{M}{M - 1} \cdot Q^A \quad (3.13)$$

ただし、

$$A = \frac{\ln M}{\ln \lambda} \quad (3.14)$$

Aの絶対値は3次元樹木形状Tのフラクタル次元を表している。よって式(3.13)(3.14)より、形状粗さQを大きくするに従って、Qの、樹木形状のフラクタル次元乗に反比例して、形状データの量を減少させられることが分かる。

### 3.4.2 簡略化形状の多重化による形状演算時間の短縮

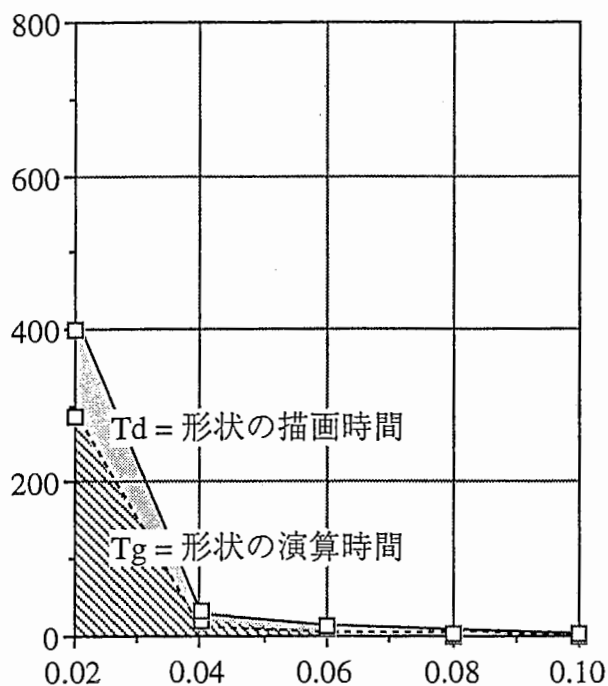
3次元樹木形状をフラクタルを用いて簡略化することで、簡略化された形状のデータ量は、形状粗さのフラクタル次元乗に反比例して減少することを述べた。しかし、形状粗さが変化する度に、フラクタルモデルにより簡略化形状を演算するのは、画像生成時間の観点から効率が悪いといえる。例えば、指定する形状粗さが元の形状と比べてあまり差のない場合には、形状生成のための縮小写像の計算時間のオーバーヘッドが無視できないほどに増大すると考えられる。ここでは、そのようなオーバーヘッドを軽減するために、描画に前もって形状粗さに応じて階層化された樹木形状表現を生成しておき、形状粗さが指定されたときにその形状表現から高速に樹木形状を生成する方法について述べる。

#### (1) 樹木画像生成時間の問題

樹木画像生成時間 $T_{total}$ は形状データの生成時間(フラクタルの演算時間) $T_g$ と描画時間 $T_d$ の合計である。図3.10に、形状粗さQを変えたときの $T_{total}$ の変化と、 $T_{total}$ 中の $T_g$ と $T_d$ の割合の例を示す。

図3.10から、樹木画像生成時間においてフラクタルの演算時間 $T_g$ が支配的であることが分かる。 $T_g$ と $T_d$ はフラクタルの演算によって生成される形状データ量に比例するので、この傾向はQの大小によらない。多数の樹木の存在するシーンを表示するために、各樹木形状のQを大きく設定しても、計算機のパワーの大部分が形状生成に費やされることになり、非常に効率が悪い。

また、Qが大きいときには $T_d$ も $T_g$ も小さくなるため、 $T_{total}$ の短縮効果が得られるが、Qが小さいときには、元の形状データをそのまま用いて画像生成処理を行った場合よりも、かえって処理時間が増大することがあり得る。

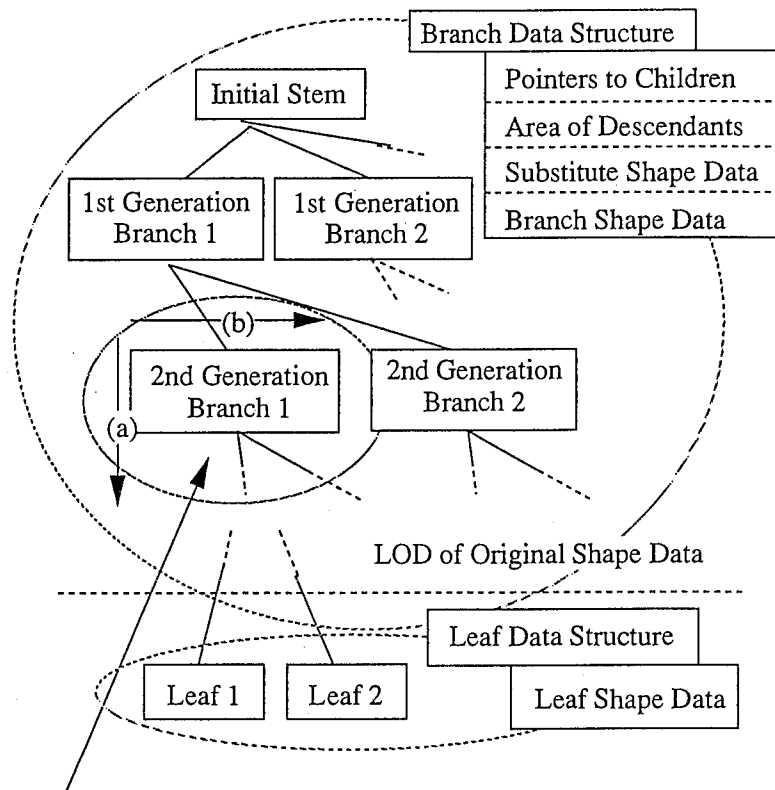


形状粗さ (簡略化形状サイズ/樹木サイズ)

図 3. 10 画像生成時間におけるフラクタル演算の占める割合

## (2) 樹木の簡略化形状の多重化

上記の問題を解決するために、フラクタルの演算時間を短縮する、形状粗さに応じた簡略化形状の多重化表現を用いる。図 3. 11 に、形状粗さに応じた樹木の簡略化形状の多重化表現で用いる具体的なデータ構造を示す。図中の各ノードは、枝データあるいは葉データを表す。各ノード間のリンクは形状データ間の親子関係を表しており、下位の形状は上位の形状から派生することを示す。枝データは、(a) 下位のデータへのポインタ、(b) 全ての下位のデータ (子孫の枝のグループ) の存在する範囲、(c) 子孫の枝のグループを置き換える単純な形状 (簡略化形状)、そして (d) 枝の形状データを属性として持つ。また葉データは、葉の形状データを属性として持つ。



Compare the area of descendants with the level of detail by using eq.(3.9').

- (a) If eq.(3.9') is false, go down to next generation nodes.
- (b) If eq.(3.9') is true, go to brother node

図 3. 1 1 簡略化形状の多重化表現のデータ構造

ここでは、子孫の枝のグループの存在範囲には図 3. 9 の  $A_1'$  の長さ  $|A_1'|$  を用いた。枝形状としては六角柱を用いた。また、葉の形状および簡略化形状には、図 3. 1 2 に示すように、レンダリングに用いる 2 次元樹木画像の樹木領域を包むポリゴン  $P$  を、 $A_1'$  を生成する操作の列  $\omega_{i \cdots \omega_1}$  で変換した、 $\omega_{i \cdots \omega_1}(P)$  を用いる。すなわち、簡略化形状の姿勢と大きさは、 $\omega_{i \cdots \omega_1}$  及び  $P$  によって定まる。3 次元樹木形状  $T$  は、例えば、枝  $A$  の世代までは枝形状を用いて、また、それ以降の世代の枝や葉はそれらを 1 つにまとめたポリゴン  $\omega_{i \cdots \omega_1}(P)$  で表示される。

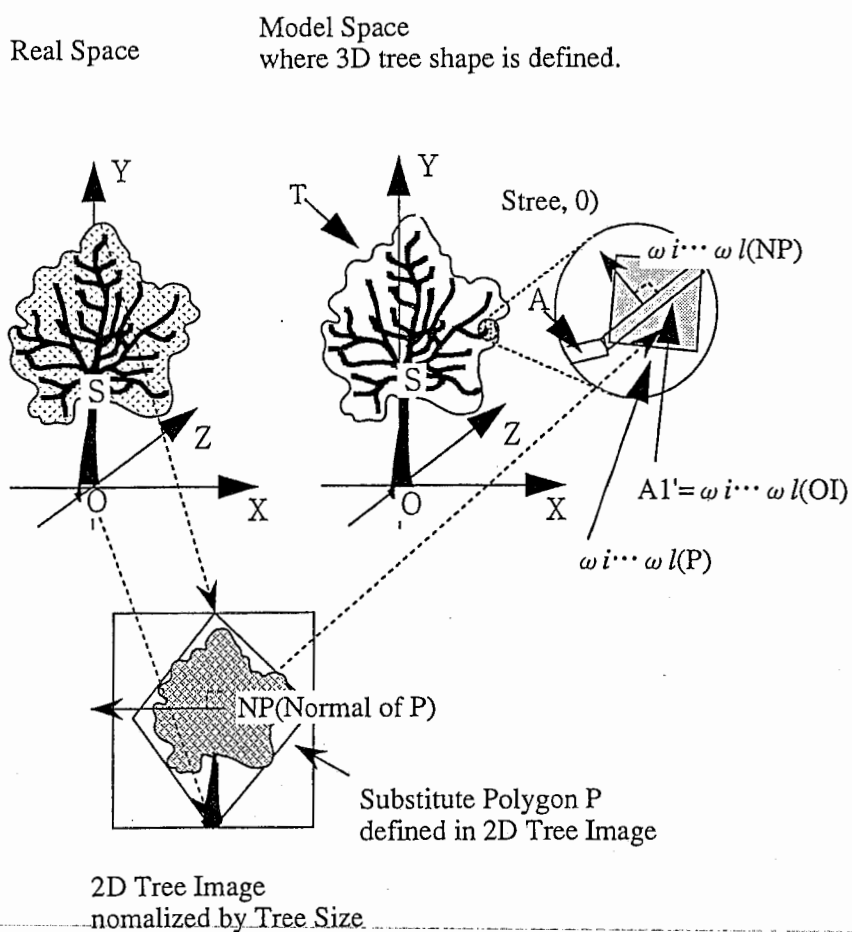


図 3. 1 2 詳細な形状を置き換える形状の生成方法

### (3) 樹木形状生成

形状粗さが指定されたときには、このデータ表現を辿って、形状粗さと各枝で計算された  $|A1'|$  を順次比較することで、樹木の簡略化形状のデータを生成する。指定した形状粗さに応じた樹木形状生成の考え方を以下に述べる。階層化された形状表現を初期値の幹から辿り、各枝で計算された  $|A1'|$  が指定する粗さより大きい場合は枝形状をコピーし、さらに下位の枝について比較を継続する (図 3. 1 1 (a))。また各枝で計算された  $|A1'|$  が指定する粗さより小さい場合には簡略化形状をコピーし、下位の枝についての比較は継続しない (図 3. 1 1 (b))。よって、与えられた形状粗さに応じた形状データは、生成されるデータの個数と等しい回数的大小比較で生成できる。

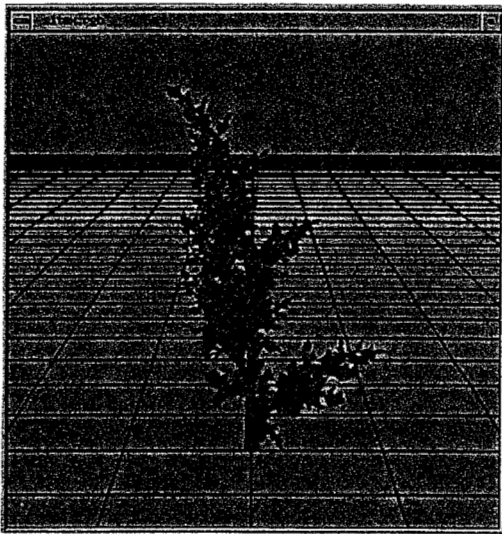


### 3. 5 実験結果および考察

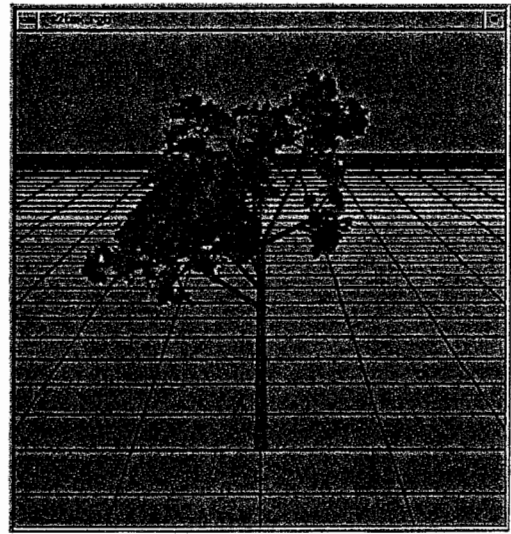
本節では、フラクタルを用いた樹木形状の簡略化による高速3次元樹木画像生成方法の有効性を確認するために、簡略化形状の形状粗さに対する樹木形状データの量、および画像生成時間についての評価をしたので、それら評価結果について述べる。

#### 3. 5. 1 形状粗さに対する簡略化形状の形状データの量

複雑な樹木形状に対して、実時間で3次元樹木画像を生成するためには、樹木形状生成時に適当な形状粗さの値を指定し、形状を簡略化する必要がある。ここでは、指定する形状粗さと生成される形状データ量の関係、及び指定する形状粗さと画像生成時間について評価する実験を行なった。実験には図3. 13に示したSample3.1, Sample3.2の2種類の樹木形状を使用した。樹木の表示



Sample3.1



Sample3.2

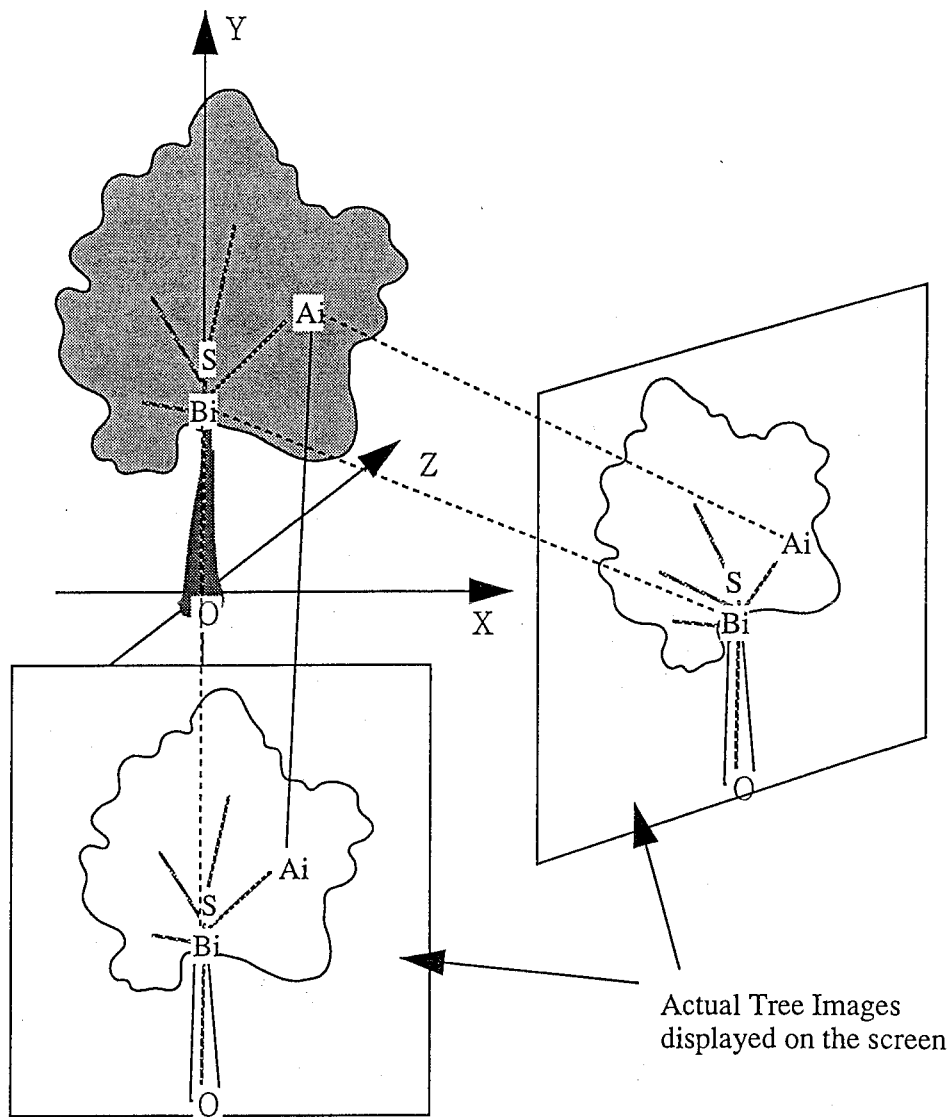
図3. 13 評価実験に用いたフラクタルモデルによる樹木形状表示例

サイズについては、スクリーンから50cm離れた視点位置から見て、表示の際の画角を $25^\circ$ とした。形状の粗さ $Q$ は、0.02から0.10の間を0.02刻みで変化させた。

各樹木形状のIFSの詳細については表3.1に示す。これらのフラクタルモデルは、実際の樹木に似せた形状を生成するように、多視点実写樹木画像を基に対話的に得た。すなわち、図3.14に示すように、複数の異なる視点からの実写樹木画像から枝 $A_iB_i$ を手動で定め、その結果を元にパラメタを計算した。そのパラメタを用いて樹木形状を生成し、元にした樹木画像と比較して、パラメタの妥当性を確認する。

表3.1 評価実験に使用したフラクタルモデルのパラメタ

	$y_i$	$s_i$	$\theta_{z_i}(\text{deg})$	$\theta_{y_i}(\text{deg})$	$\lambda$	$\overline{OS}$	$M$
Sample3.1	0.40	0.60	10.0	0.0	0.46	0.4	5
	0.24	0.55	40.0	60.0			
	0.16	0.45	-40.0	120.0			
	0.08	0.40	50.0	180.0			
	0.08	0.30	-60.0	50.0			
Sample3.2	0.52	0.52	-32.6	40.2	0.45	0.52	6
	0.44	0.52	33.9	40.5			
	0.44	0.38	7.3	45.0			
	0.40	0.47	56.4	45.2			
	0.52	0.28	62.8	180.0			
	0.36	0.55	-59.3	134.8			



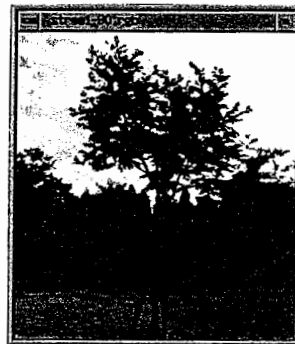
- (a)  $\overline{AiBi}$  can be moved interactively to determine its parameters.
- (b) The tree shape can be generated by using all parameters determined, and can be compared with the actual tree shapes in their images.

図 3. 1 4 フラクタルモデルの縮小写像のパラメタの編集

図3.15に、多視点実写樹木画像の例と、それをもとに編集を行なって得られたパラメタにより生成した3次元形状例を示す。実写樹木画像は正面と、左回りに90度視点移動した側面より撮影して得られたものである。



Front View

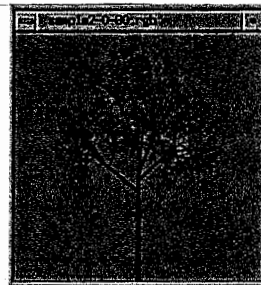


Side View

Multi-viewpoint Actual Tree Images



Front View



Side View

Obtained 3D Tree Shape Data

図3.15 多視点樹木画像と編集されたフラクタルモデルによる3次元樹木形状

図 3. 1 6 に、指定した形状粗さ  $Q$  に対する生成された形状データの量  $N$  を示す。この実験結果を、式 (3. 1 3) と対応させると、次のような実験式を得る。

$$N = e^{-0.22} \cdot Q^{-2.18}, \quad \text{sample3.1} \quad (3. 1 5)$$

$$N = e^{0.17} \cdot Q^{-2.39}, \quad \text{sample3.2} \quad (3. 1 6)$$

2 種類の樹木モデルは共に、形状粗さ  $Q$  に対して、 $Q$  のフラクタル次元 ( $> 2$ ) 乗に反比例してデータ量  $N$  が減少する。一方、指定された分解能以下の形状の凸凹を除去する方法では、形状粗さ  $Q$  に対して、 $Q$  の 2 乗以下に反比例してデータ量  $N$  が減少するので、フラクタルによる形状簡略化のほうがデータ量削減効果が高いといえる。また、樹木形状のデータ量が多いということは、樹木形状が複雑である、すなわち、樹木形状のフラクタル次元が高いことを意味している。よって、複雑なデータ量の多い樹木形状ほど、形状簡略化によるデータ量削減効果が高いことを示している。

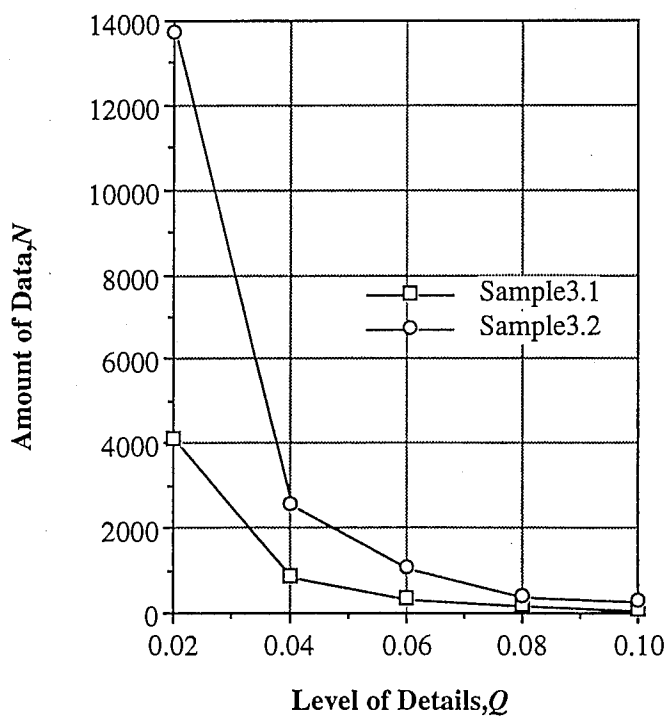


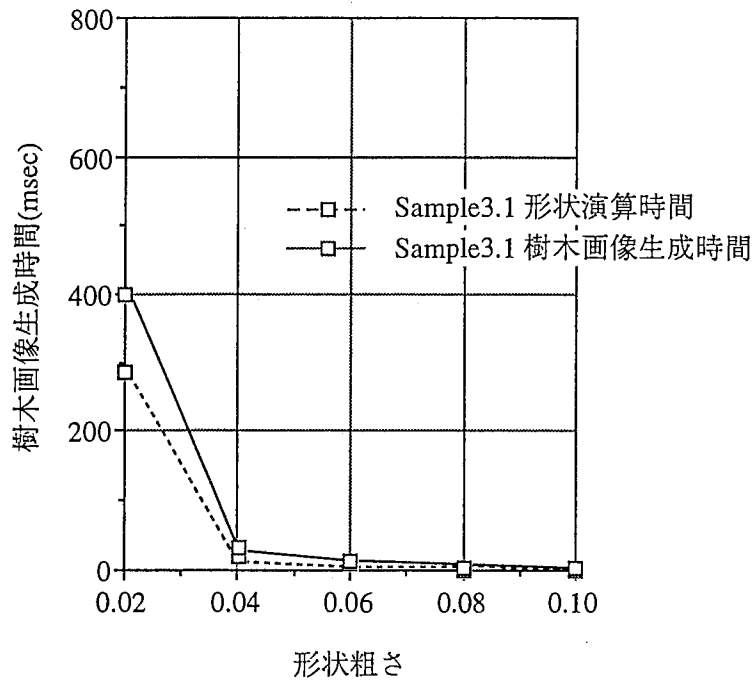
図 3. 1 6 形状粗さに対する簡略化形状のデータ量

### 3. 5. 2 形状粗さに対する 3次元樹木画像生成時間

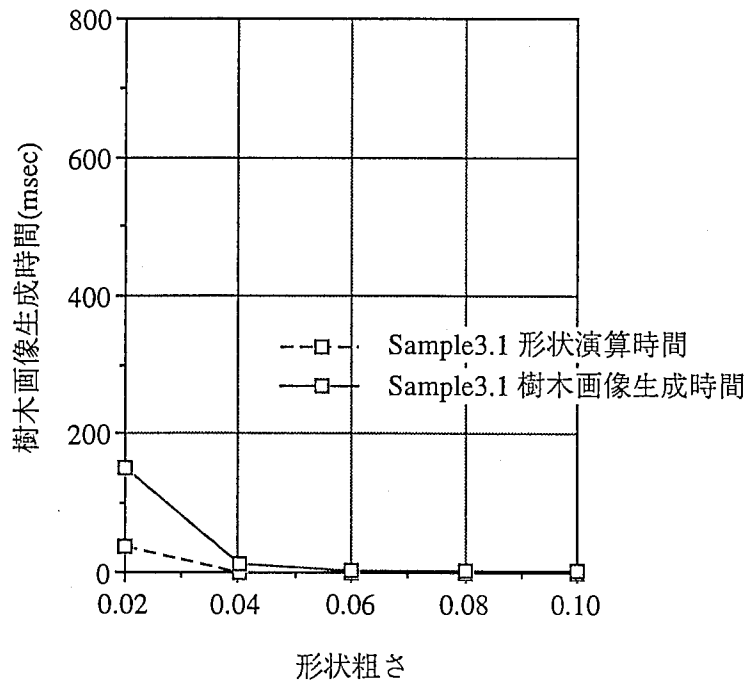
図3. 17に、Sample3.1を用いた場合の、与えた形状粗さQに対する、樹木形状を計算する時間と、最終的に描画する時間についての実験結果を示す。実験の条件は、3. 5. 1と同じである。簡略化された樹木形状を表示するために、5章で述べる樹木画像のテクスチャマップによるレンダリング、光源計算、グーロシェーディングを用いた。画像生成時間の測定の実験には、Silicon Graphics社のCrimson RE（性能110万Tmesh/秒）を用いた。ただし、この性能は面積50ピクセルの3角形メッシュを用い、テクスチャマップ有、光源計算なし、フラットシェーディングなどの条件下で測定されたものである。描画に使用した樹木形状の、最小のプリミティブは4角形ポリゴンであり、本実験に使用した装置では、これは2つの3角形メッシュに分割して処理される。

図3. 17(a)には、簡略化形状の多重化を用いずに画像生成を行った場合を、また図3. 17(b)には、簡略化形状の多重化を用いて画像生成を行った場合の結果を示す。図3. 17(a)では、 $Q=0.02$ のときのフラクタルの演算時間は、約280msecである。一方、図3. 17(b)では、その演算時間は約40msecとなることから、簡略化形状の多重化を用いることで、フラクタルの演算時間が約1/7に短縮されることが分かる。

図3. 18には、Sample3.2を用いた場合の、与えた形状粗さQに対する、樹木形状を計算する時間と、最終的に描画する時間についての実験結果を示す。図3. 17(b)の場合と同様に、与えた形状粗さQが大きくなり形状データ量が減少するにつれて、画像生成時間が短縮されていくことが分かる。また、フラクタルの演算時間の占める割合も、図3. 17(b)と同程度といえる。



(a) 簡略化形状の多重化を用いなかった場合



(b) 簡略化形状の多重化を用いた場合

図 3. 17 形状粗さに対する樹木画像生成時間 (1)

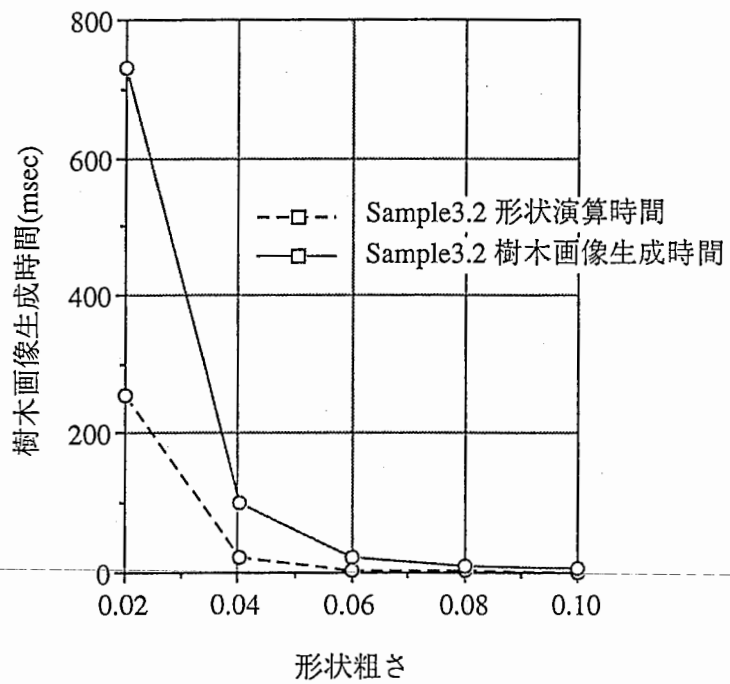


図 3. 1 8 形状粗さに対する樹木画像生成時間 (2)



つぎに、図 3. 1 9 に、Sample3.1, Sample3.2 の 2 種類の樹木を用い、それらの向き、大きさを変えて、それぞれ 15 本をシーン中に配置し、フラクタルを用いて形状簡略化を行った画像の生成例を示す。図 3. 1 9 では、表示画面上の簡略化形状のサイズが一定となるように、形状粗さを制御した。その結果、フラクタルによる形状の簡略化を実施しない場合に約 270,000 ポリゴン/シーンが、フラクタルによる形状簡略化によって、1 / 1 4 の約 16,000 ポリゴン/シーンに減少した。

画像生成時間の測定の実験には、前述の実験と同様に、Silicon Graphics 社の Crimson RE を用いた。このときの画像生成時間は、315msec（形状演算時間 53 msec, 描画時間 262 msec）であった。また、Sample3.1 の簡略化形状の多重化表現のデータサイズは約 98K byte, テクスチャデータサイズは約 4 M byte, Sample3.2 の簡略化形状の多重化表現のデータサイズは約 340K byte, テクスチャデータサイズは約 4 M byte である。

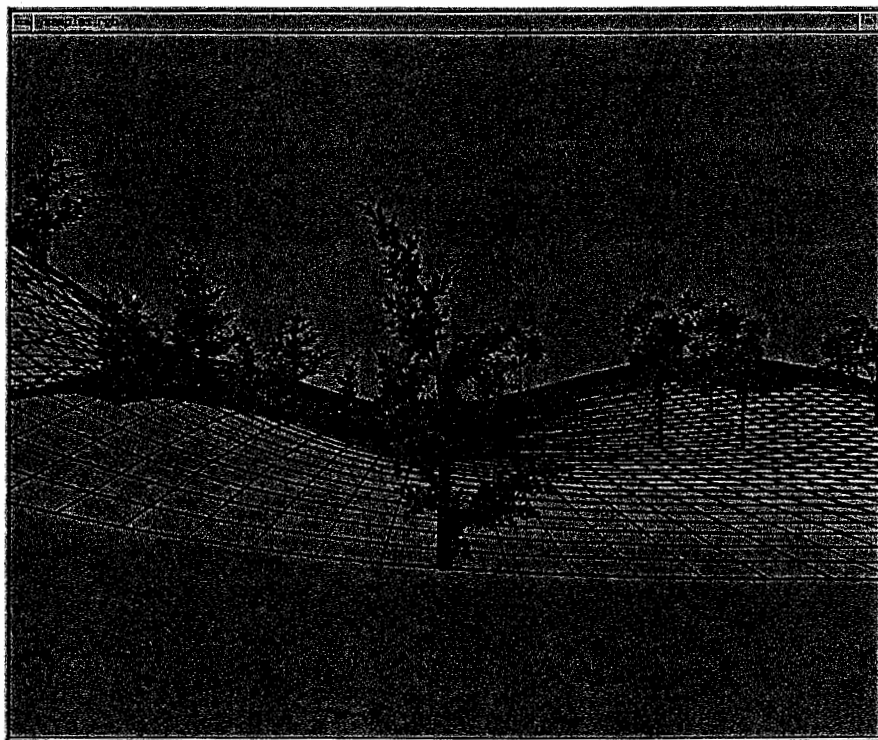


図 3. 1 9 多数の樹木の存在するシーンの表示例

### 3.6 まとめ

本章では、フラクタルの性質を利用することで、樹木のような複雑形状が効果的に簡略化できることを示した。

つぎに、フラクタルによる簡略化形状の演算時間を短縮するための、簡略化形状の多重化方法を提案した。

そして、フラクタルを用いた樹木形状簡略化と簡略化形状の多重化方法の評価として、指定する形状粗さに対する3次元樹木形状データの量、及び画像生成時間について、実際に計算機上で実験した結果を示した。以下に実験の結果および考察をまとめる。

(1) 実験結果から、フラクタルを用いて樹木形状を簡略化することで、樹木形状データ量は形状粗さに対し、樹木形状のフラクタル次元乗に反比例して減少することを示した。

(2) 従来の指定された分解能以下の形状の凸凹を除去する方法は、形状粗さの2乗以下に反比例して形状データ量が減少することから、提案した手法は、従来手法に比べてデータ量削減効果が高いといえる。

(3) 樹木形状データ量は形状粗さに対し、樹木形状のフラクタル次元乗に反比例して減少することから、フラクタル次元の高い、すなわち複雑なデータ量の多い樹木形状ほど、形状簡略化によるデータ量削減効果が高いといえる。

(4) 簡略化形状の多重化により、画像生成時の形状演算時間を大幅に短縮できることを確認した(約1/7)。

## 4. 樹木形状の再構成技術 [53, 54]

### 4. 1 まえがき

仮想空間中に表示する3次元樹木形状を生成する方法として、従来の成長モデルに基づく樹木形状生成手法では、成長のパラメタを設定してシミュレーションを行ない、その結果を確認し、更にパラメタを調整するといった、対話的な作業が必要である。しかし、目に留まった絵画や写真中の樹木形状が欲しいときには、その画像を入力として、意図する3次元樹木形状を生成するパラメタが、自動的に推定できることが望ましい。これまでに、3次元形状の任意方向からのシルエット画像を用いた形状再構成方法が研究されているが、樹木のように、シルエット内部に微細で入り組んだ構造を持つ形状に対しては有効でないことは、先に述べた。そこで、3次元樹木形状は統計的にフラクタル性を有すること、そして、3章で述べたように、高速樹木画像生成のためには、樹木形状のフラクタルモデルが必要であることから、本章では、3次元樹木形状の正面図と側面図を入力として、意図する3次元形状を生成するフラクタルモデルを、自動的に推定する方法を述べる。

今回提案する手法は、樹木画像から意図する3次元樹木形状を自動的に生成するために、3次元樹木形状の正面と側面の2枚のシルエット画像を入力として、それによく適合する形状を生成するよう、フラクタルモデルのパラメタを推定する。今回は、入力画像が必要最小限の場合を評価するため、正面と側面の2枚のシルエット画像を入力とした。

### 4. 2 コラージュ定理と形状再構成に使用したフラクタルモデル

本節では、フラクタル形状の再構成に関して重要である、コラージュ定理について述べ、つぎにコラージュ定理に基づき、シルエット画像からの3次元樹木形状の再構成に用いるフラクタルモデルの、縮小写像の個数について考察した結果を述べる。

#### 4. 2. 1 コラージュ定理 [49-52]

##### (1) ハウストルフ距離

空間R中の空でない2つの部分集合A, Bがどの程度一致しない集合であるかを計量するための量として、以下のようにハウストルフ距離が定義されている。

まず、Aの $\varepsilon$ -カラー [52] というものを、式(4.1)のように定義する(図4.1)。

$$A\varepsilon = \{x \in A \mid \exists y \in A \Rightarrow d(x, y) \leq \varepsilon\} \quad (4.1)$$

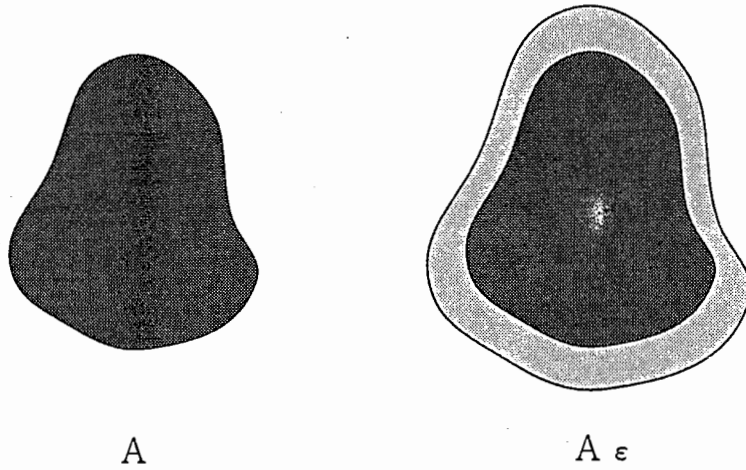


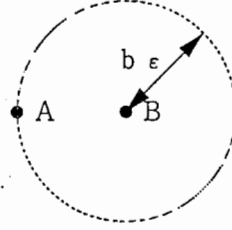
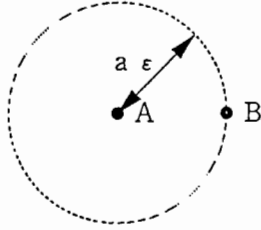
図 4. 1  $\epsilon$  - カラーの例

B の  $\epsilon$  - カラーについても同様に定義する。このとき A と B のハウスドルフ距離  $h$  は、式 (4. 2) のように定義される。

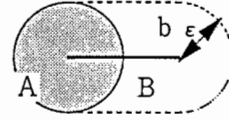
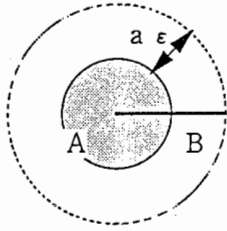
$$h(A, B) = \inf\{\epsilon \mid (A \subset B\epsilon) \text{ and } (B \subset A\epsilon)\} \quad (4. 2)$$

図 4. 2 (a) ~ (d) にハウスドルフ距離の例を示す。いずれの例でも、2次元の距離空間上の2つの部分集合 A, B に対して、図 4. 2 の左側に、 $a_\epsilon = \inf\{\epsilon \mid B \subset A\epsilon\}$ 、図 4. 2 の右側に、 $b_\epsilon = \inf\{\epsilon \mid A \subset B\epsilon\}$  を示した。このとき、ハウスドルフ距離は、 $h(A, B) = \max\{a_\epsilon, b_\epsilon\}$  となる。(a) は A, B とも点の場合で、このときのハウスドルフ距離はユークリッド距離に一致する。(b) は A が円領域で B が線分の場合、(c) は A が円領域で、正方形領域の B に含まれる場合、(d) は A, B ともに円領域で、一部が交わっている場合である。

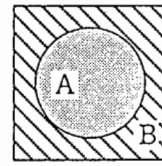
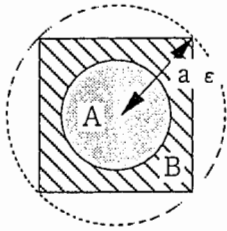
(a)



(b)



(c)



(d)

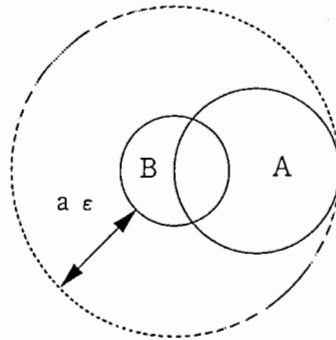
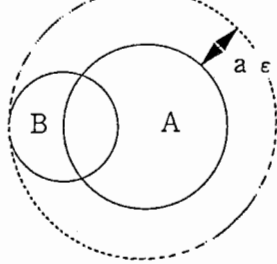


図 4. 2 ハウスドルフ距離の例

(2) コラージュ定理

フラクタル形状の再構成に対しては、コラージュ定理がひとつの指針を与えている。すなわち、ターゲット  $T$  を適当な 3 次元形状、アトラクタ  $A$  を 3 次元フラクタル形状、 $\{\omega_1, \dots, \omega_N\}$  を  $A$  の IFS、 $\epsilon (>0)$  を適当な実数としたとき、

$$h(T, \bigcup_{n=1}^N \omega_n(T)) < \epsilon \Rightarrow h(T, A) < \frac{\epsilon}{1-s} \quad (4.3)$$

が成り立つ。但し、 $h$  はハウスドルフ距離、 $s$  はリプシッツ数を表す。式 (4.3) は、 $\epsilon$  を小さくすること、すなわち、 $T$  を  $\bigcup \omega_i(T)$  でほとんど覆うことができれば、縮小写像を再帰的に適用して得られる  $A$  は、 $T$  を良好に近似したものになることを示したものである。図 4.3 では、葉の形状をターゲットにしたとき、その縮小コピーによりターゲットが十分に覆われた場合 (図 4.3 上) と、そうでない場合 (図 4.3 下) を示した。図 4.3 に右側には、それぞれの場合で生成されたフラクタル形状を示す。ターゲットの縮小コピーによりターゲットが十分に覆われた場合には、ターゲットと生成されるフラクタル形状はよく一致しているが、そうでない場合には、ターゲットと全く異なる形状が生成される。以上のように、式 (4.3) の  $\epsilon$  を小さくするように縮小コピーを配置するよう、縮小写像の各パラメタを制御することで、フラクタル形状の再構成が可能となることがわかる。

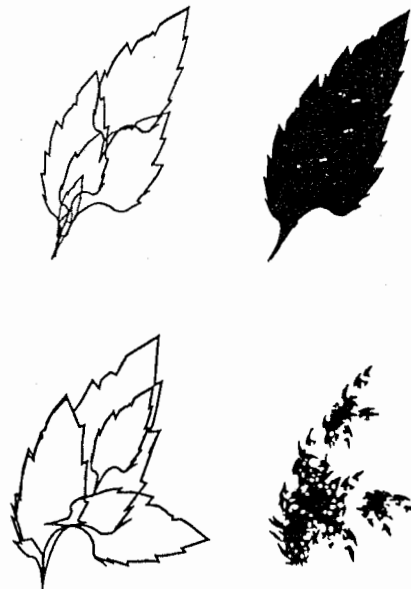


図 4.3 コラージュ定理の例

#### 4. 2. 2 形状再構成に使用したフラクタルモデル

本章で提案するフラクタルモデルの再構成において使用する、フラクタルモデルを構成する縮小写像  $\omega_i$  ( $1 \leq i \leq n$ ) の形式は、3章で用いたものと同様、以下のように記述される (式 4. 4) (図 4. 4).

$$\omega_i = \text{Trans}(0, y_i, 0) \cdot \text{Roty}(\theta_{y_i}) \cdot \text{Rotz}(-\theta_{z_i}) \cdot \text{Scale}(s_i) \quad (4. 4)$$

ただし、

$\text{Trans}(x, y, z)$	:	平行移動
$\text{Rotz}(\theta)$	:	z 軸周りの回転
$\text{Roty}(\theta)$	:	y 軸周りの回転
$\text{Scale}(s)$	:	拡大縮小

また、 $y_i$  は枝  $A_i B_i$  が発生する位置、 $s_i$  は幹  $OS$  に対する枝  $A_i B_i$  の長さ比、 $\theta_{z_i}$ 、 $\theta_{y_i}$  はそれぞれ、枝  $A_i B_i$  の延びる方向の z 軸周り、y 軸周りの回転量を表す。

上記の樹木形状のフラクタルモデルに対し、縮小写像の個数を増してモデルの自由度を高めることで、シルエットと、再構成される形状が良く一致するようになることは予想されるが、パラメタの個数が多くなることでパラメタの推定は困難になる。実際、 $T$  を単位円、また各  $\omega_n(T)$  を大き

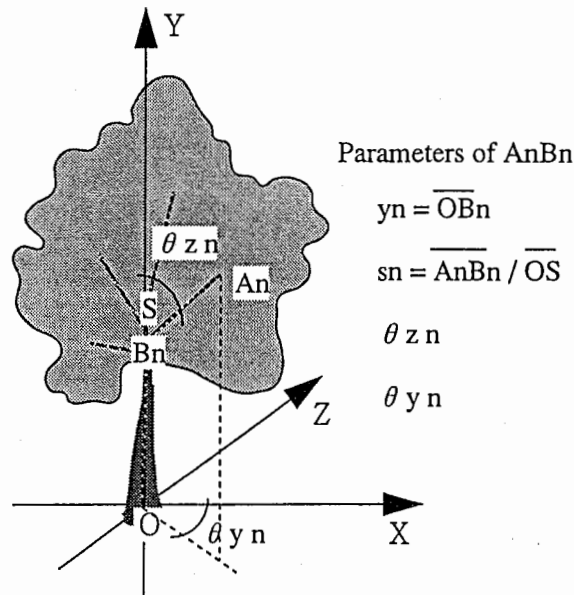


図 4. 4 フラクタルモデルを構成する縮小写像のパラメタ

さの等しい円とした単純な場合を考えたとき、図4.5に示すように、縮小写像の個数を増やすと  $h(T, \cup_{n=1}^N \omega_n(T)) / (1-s)$  の上限の値は減少する。特に、 $\omega_n(T)$  の個数が2と3の間でそれが顕著である。Tが任意の形状であってもこの傾向は同様であると予想されるので、1つのシルエットを3つの縮小コピーで覆うのが最も効率が良いと考えられる。そこで図4.6(1)のように、各シルエットに3つの縮小写像  $S_{stem}$ ,  $S_{branchi}$ ,  $S_{branchi+1}$  を割り当てる。各シルエットに割り当てた縮小写像のうち、 $S_{stem}$  はシルエット間で共通とするのが自然である。また今回、入力シルエット画像は正面図と側面図であり、正面図では、 $S_{branchi}$  ( $i=3,4$ ) による縮小コピーは、 $S_{stem}$ ,  $S_{branchi}$  ( $i=1,2$ ) による縮小コピーの内部にほとんど含まれる。すなわち正面図においては、 $S_{branchi}$  ( $i=1,2$ ) がシルエット画像との適合度に対して支配的であり、またその逆も言える。よって、 $S_{branchi}$  ( $i=1,2$ ) と  $S_{branchi}$  ( $i=3,4$ ) は、個別に制御可能なことが望ましいと考え、 $S_{branchi}$  ( $1 \leq i \leq 4$ ) は各シルエットで独立とした。すなわち図4.6(2)に示すように、今回、3次元樹木形状の近似に用いるフラクタルモデルの、縮小写像の個数は5個とする。

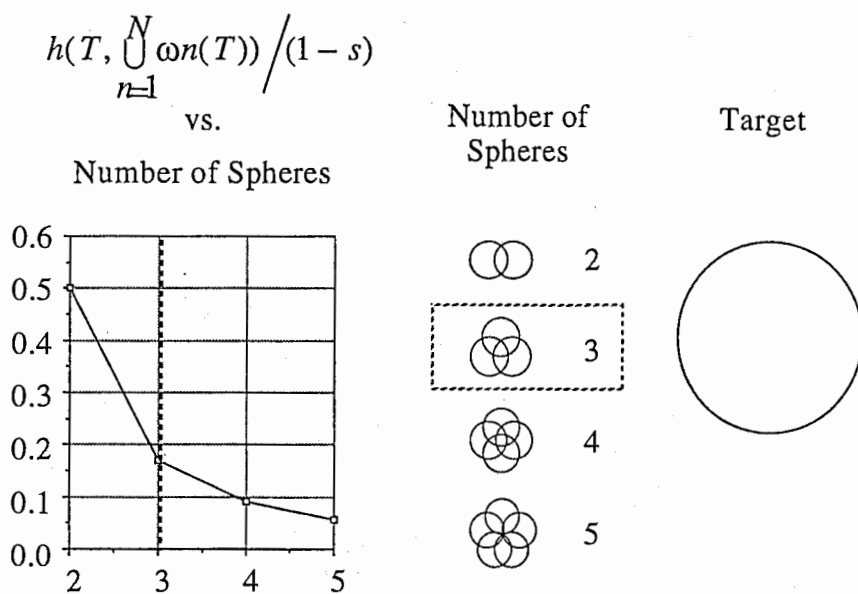
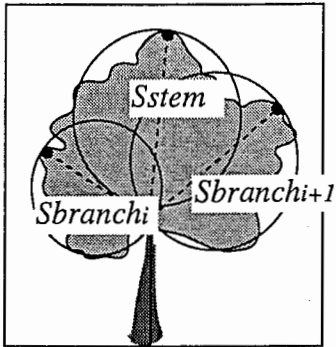


図4.5 シルエットと生成されるフラクタル図形間のハウスドルフ距離の評価

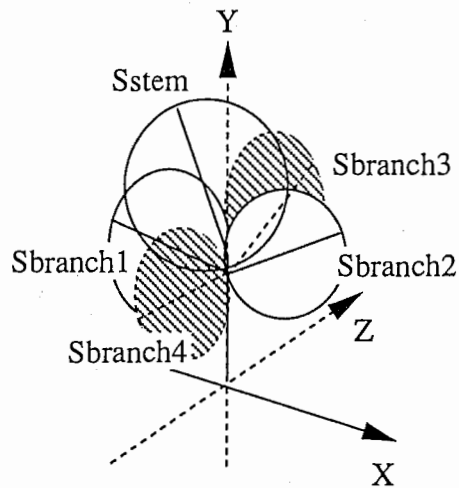


Front and Side View



*Sstem* :Contraction map of Stem  
*Sbranchi* :Contraction map of left Branch  
*Sbranchi+1* :Contraction map of right Branch

(1) Contraction maps allocated to each Silhouette



(2) Fractal Model of 3D Tree for Parameter Estimation

図 4 . 6 樹木の正面図と側面図に形状を適合させるのに用いたフラクタルモデル

#### 4. 3 多視点樹木画像を用いたフラクタルモデルの推定方法の概要

本節では、図4.7に示すように、フラクタルモデルのパラメタを調節して、3次元樹木形状の正面図と側面図にフラクタル形状を合わせ込み、仮想空間の3DCG表示に使用するフラクタルモデルを得る方法について、その概要を述べる。

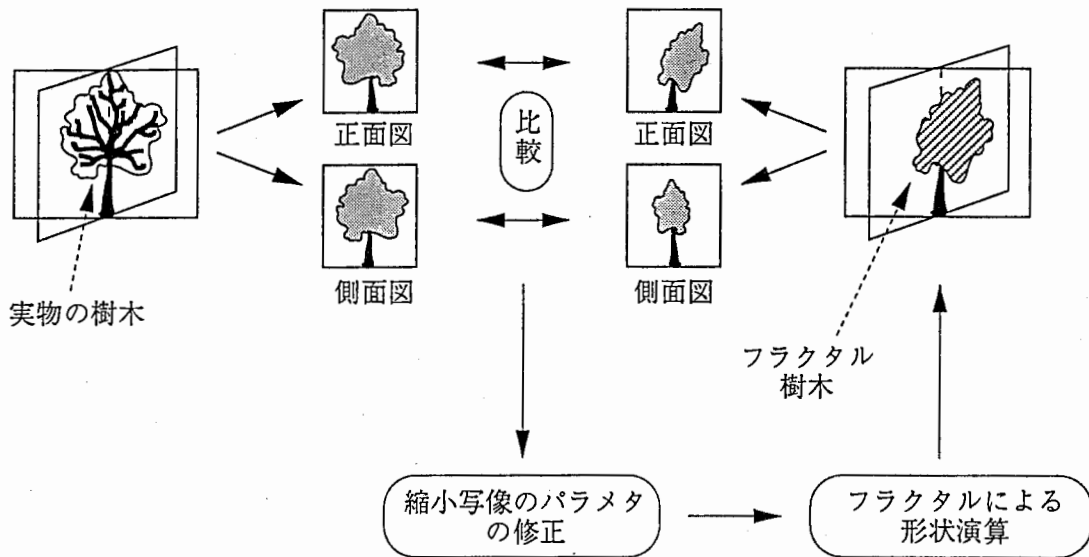
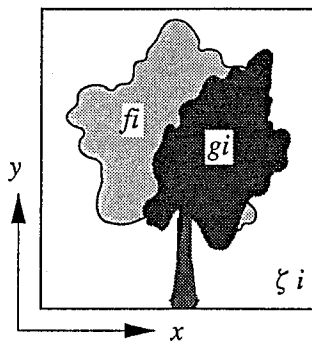


図4.7 多視点樹木画像を用いた3次元樹木形状を近似するフラクタルモデルの推定

式(4.3)に示したコラージュ定理は、ターゲットの形状 $T$ の $\{\omega_1, \dots, \omega_N\}$ による縮小コピーが、 $T$ 全体を効率良く包むとき、 $\{\omega_1, \dots, \omega_N\}$ は $T$ に近いフラクタル形状 $A$ を生成することを示している。例えばフラクタル符号化では、オリジナルの画表面 $T$ と、 $\omega_n(T)$ の和集合が成す画表面の差を小さくする $\{\omega_1, \dots, \omega_N\}$ を推定する。しかしここでは樹木の正面図と側面図が入力であり、 $T$ が未知のため、式(4.3)を用いることができない。また、実時間性が考慮されるべき応用では、式(4.3)の右辺の評価は、各縮小写像の適用が1回ですむ利点があるが、本手法は、CGによる3次元樹木の実時間表示の前処理として用いるものであり、実時間性は考慮する必要はない。そこで本手法では、 $\{\omega_1, \dots, \omega_N\}$ によりフラクタル形状 $A$ を生成し、その投影画像と、樹木の正面、側面のシルエットとを直接比較することにする。

コラージュ定理では、ターゲットの形状Tと生成されるフラクタル形状Aの適合度の評価尺度に、ハウスドルフ距離を用いている。しかし実用的には、ハウスドルフ距離の代わりに、例えばフラクタル符号化では、オリジナルの画表面Tと、 $\omega_n(T)$  の和集合が成す画表面の間の、各画素値の差分のRMS (Root Mean Square) を用いている。本手法では、Aの投影画像と樹木の正面、側面のシルエットとの比較の評価尺度として、式(4.5)に示すように、各シルエットの正規化された相互相関 [116] の和を用いることにする。式(4.5)では、 $f_i$ をフラクタル形状Aの投影画像、 $g_i$ をシルエット画像、また $\zeta_i$ を $f_i$ を含む領域を示す( $i$ =正面図, 側面図)(図4.8)。式(4.5)は、 $g_i = c \cdot f_i$ のとき最大になる。

$$H(\omega_1, \dots, \omega_5) = \sum_i \frac{\iint_{\zeta_i} f_i(x, y) g_i(x+u, y+v) dx dy}{\sqrt{\iint_{\zeta_i} f_i^2(x, y) dx dy \iint_{\zeta_i} g_i^2(x+u, y+v) dx dy}} \quad (4.5)$$



- $\omega_n (n=1...5)$  フラクタルモデルの縮小写像 (4パラメタ×5)
- $f_i$  実写樹木画像 (シルエット画像)
- $g_i$  フラクタル樹木のシルエット画像
- $\zeta_i$  シルエットを比較する範囲
- $i$  正面図, あるいは側面図

図4.8 シルエット画像とフラクタル形状の投影画像との相互相関の計算

ところで、4.2で述べたように各 $\omega_n$ は4つの独立なパラメタを有し、また、 $H(\omega_1 \cdots \omega_5)$ は非線形であるので、式(4.5)は $4 \times 5$ 次元の非線形最適化問題である。ここでは最適傾斜法を用いて、 $H(\omega_1 \cdots \omega_5)$ を最大にする $\{\omega_1, \dots, \omega_5\}$ を求める。最適傾斜法で得られる解は局所的な最適解であり、初期値の与え方によって解の精度が異なる。そのため探索の途中に散乱を加えて精度の高い解を求めるなどの工夫が必要となる。ここでは、正面、及び側面のシルエットを比較的良く近似する点を簡易な手法で求め、これを最適傾斜法の解の探索開始点とすることで、最適傾斜法で得られる解の精度を高めることにした。

## 4. 4 最適傾斜法による縮小写像のパラメタ推定

与えられた目的関数 $f(x)$ を最小(最大)にする変数 $x$ を求める際に、それを解析的に解くことが不可能な場合でも、線形計画法、あるいは非線形計画法を用いることで、近似的に $x$ を求めることができる。ここでは、式(4.5)が非線形であることから、非線形計画法の1つである、最適傾斜法を用いることにした。本節では、最適傾斜法について簡単に説明した後、最適傾斜法を用いて、式(4.5)を最大にするように縮小写像のパラメタを推定する方法について述べる。

### 4. 4. 1 最適傾斜法について

目的関数 $f(x)$ の傾斜が求められるとき、常に曲面の最大の傾斜方向に向かっていくことで局所的な最適解に到達する、最大傾斜法と呼ばれる方法が存在する。例えば $f(x)$ を最大にする $x^*$ を求める場合、 $f(x) = const$ の等高線の法線方向に添って下っていく。このためには、

$$x^{(m+1)} = x^{(m)} + k \cdot g^{(m)} \quad (k > 0) \quad (4.6)$$

を、初期値 $x^{(0)}$ からスタートして、 $|x^{(m+1)} - x^{(m)}| < \epsilon$ となるまで、繰り返し計算を行なえばよい。ただし、

$$x^{(m)} = \begin{bmatrix} x_1^{(m)} \\ \vdots \\ x_n^{(m)} \end{bmatrix}, \quad g^{(m)} = \begin{bmatrix} df/dx_1 \\ \vdots \\ df/dx_n \end{bmatrix} \quad (4.7)$$

しかし、この方法では、 $k$ の値を大きく取り過ぎると発散することがあり、また、 $k$ の値を小さくすれば、収束が遅くなるという欠点があり、最適な $k$ を選ぶことが難しい。最適傾斜法はこれを改善するために、以下のようなアルゴリズムで処理を行なう。

(1)  $m=0$ とする。

(2)  $x^{(m)}$ での勾配、 $g^{(m)}$ を計算する。

(3)  $f(x^{(m)} + \alpha \cdot g^{(m)})$ を最大にする $\alpha$ を探索する。これは $\alpha$ に関する1次元の探索問題である。

(4)  $x^{(m+1)} = x^{(m)} + \alpha \cdot g^{(m)}$ とし、 $f(x^{(m+1)})$ を計算する。

- (5)  $|f(x^{(m+1)}) - f(x^{(m)})| < \epsilon$ であれば処理を終了する。そうでなければ、 $m = m+1$ として、再び(2)から処理を実行する。

以上の処理を行なうことで、図4.9に示すように、 $f(x)$ の最大値が得られる。

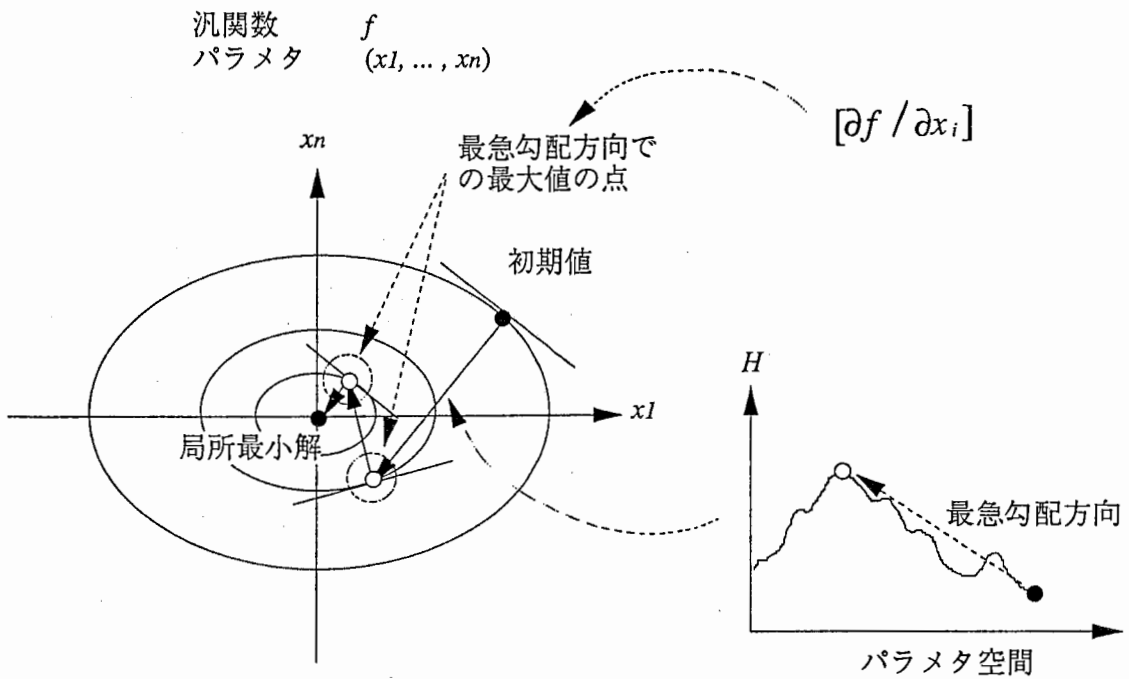


図4.9 最適傾斜法による局所最適地の推定

#### 4.4.2 最適傾斜法を用いた縮小写像のパラメタの推定方法

ここでは、先に説明した最適傾斜法を用いて、式(4.5)を最大にする縮小写像のパラメタを推定する方法について述べる。以下では、入力として $N \times N$ 画素のシルエット画像を用いる。

- (1) 縮小写像の初期値  $\{\omega^{(0)}_1, \dots, \omega^{(0)}_5\}$  を適当に定める.
- (2)  $m = 0$  とする.
- (3) 縮小写像  $\{\omega^{(m)}_1, \dots, \omega^{(m)}_5\}$  により, フラクタル形状  $A^{(m)}$  を生成する.
- (4) 式(4.5)により, そのときの各シルエット上での生成されたフラクタル形状との相互  
相関の和を計算する.
- (5) 縮小写像  $\{\omega^{(m)}_1, \dots, \omega^{(m)}_5\}$  による,  $A^{(m)}$  の縮小コピーの先端の座標値の組み  $s^{(m)}_i$   
と, 終端の座標値の組み  $e^{(m)}_i$  を求める ( $i = 1, \dots, 5$ ). このとき,  $s^{(m)}_i$  は3自由度で,  $e^{(m)}_i$   
は1自由度である.
- (6)  $\hat{x}^{(m)} = \{s^{(m)}_i, e^{(m)}_i \mid i=1, \dots, 5\}$ ,  $f(x^{(m)}) = H$ ,  $\Delta x^{(m)}_j = 0.5$  ( $j = 1, \dots, 20$ ) として, 4.  
4.1の(2)のごとく,  $x^{(m)}$  での勾配  $g^{(m)}$  を計算する. つぎに, 4.4.1の(3)(4)  
を実行し  $x^{(m+1)}$ ,  $f(x^{(m+1)})$  を求める.
- (7)  $|f(x^{(m+1)}) - f(x^{(m)})| < \epsilon$  であれば処理を終了する.
- (8)  $x^{(m+1)} = \{s^{(m+1)}_i, e^{(m+1)}_i \mid i=1, \dots, 5\}$  を用いて, 縮小写像  $\{\omega^{(m+1)}_1, \dots, \omega^{(m+1)}_5\}$  を計  
算し,  $m = m+1$  として, 再び(2)から処理を実行する.

#### 4.4.3 最適傾斜法の解の探索開始点の決定

本節では, 最適傾斜法の解の探索開始点として, 正面, 及び側面のシルエットを比較的良く近似する点を求める簡易な手法について説明する. 本手法の基本的な考え方は, 図4.10に示すように, フラクタル形状  $A$  の  $\omega_n$  による縮小コピーの投影  $PA$  を, シルエット画像のエッジ付近で適合するように, シルエット上で回転, 平行移動, 拡大縮小して  $\omega_n$  のパラメタを定める. このとき  $PA$  の先端  $E$  を  $A$  のエッジ上に, また  $PA$  の付け根にあたる  $S$  を, シルエットの幹の軸上に拘束する. これにより,  $\omega_n$  のパラメタを2自由度に制限し, 決定を容易にする.

具体的には適当に定めた  $\{\omega_1, \dots, \omega_5\}$  により生成されるフラクタル形状  $A$  の, 各  $\omega_n$  による

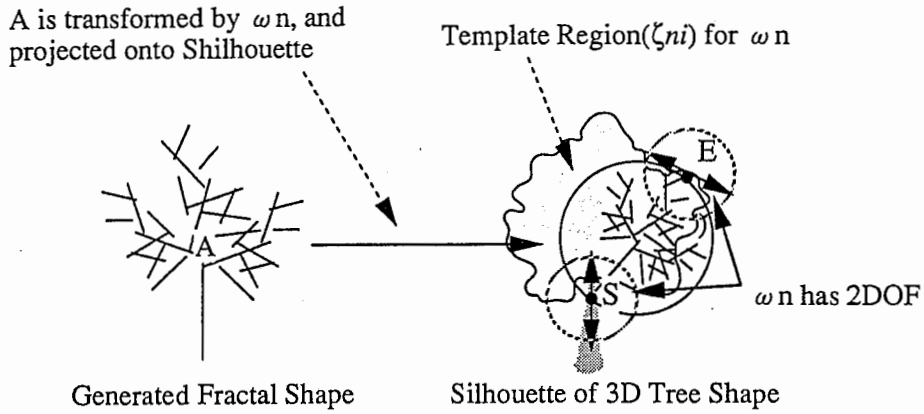


図 4. 10 パラメタの自由度を制限した  $\omega_n$  のテンプレートマッチングを用いた推定

縮小コピーを各シルエット上に投影しテンプレートとし、各シルエット上でテンプレートマッチングを実施する。テンプレートマッチングとは、ある画像パターンをテンプレートとして、解析対象の画像中で、そのテンプレートに適合する部分を見つけだすことであり、そのために、テンプレートと画像の適合の度合いを示すある種の測度を最大にする。本処理ではこの測度として、最適傾斜法と同様に、正規化された相互相関を用いる (式 (4. 8))。

$$H_{ni}(\omega_n) = \frac{\iint_{\zeta_{ni}} f_{ni}(x, y) g_i(x+u, y+v) dx dy}{\sqrt{\iint_{\zeta_{ni}} g_i^2(x+u, y+v) dx dy}} \quad (4. 8)$$

式 (4. 8) では、 $f_{ni}$  をフラクタル形状 A の  $\omega_n$  による縮小コピーの投影画像、 $g_i$  をシルエット画像、また  $\zeta_{ni}$  を  $f_{ni}$  を含む領域を示す ( $i$ =正面図, 側面図)。ここでは、 $f_{ni}$  を含む小領域  $\zeta_{ni}$  は、A の  $\omega_n$  による縮小コピーの投影画像を包む円の領域とする。テンプレートマッチングによる最適傾斜法の解の探索開始点の決定を、以下にまとめる。



- (1) 初期値とする  $\{\omega_1, \dots, \omega_5\}$  を, 正面図と側面図から定め, フラクタル形状 A を生成する. 正面図, 側面図それぞれに割り当てられた各 3 つの縮小写像による縮小コピーを円で代用し, それら円の和集合が各シルエットを包むよう  $\{\omega_1, \dots, \omega_5\}$  を定める. このとき  $\theta_{y1} = \theta_{y2} = 0$ ,  $n \geq 2$  のとき  $\theta_{yn} = 90 \times (n-2)$ , とする.
- (2) 図 4. 10 に示すように, 正面図, 側面図それぞれに割り当てられた各 3 つの縮小写像によりフラクタル形状 A を変換し, 各画像に投影したものをそれぞれテンプレートとする.
- (3)  $\theta_{yn}$  を (1) のように拘束し, また図 4. 10 に示すように, テンプレート画像の先端 E をシルエットのエッジ上に, また付け根にあたる S をシルエットの幹の軸上に拘束のもとで, 各  $H_i(\omega_n)$  を最大にするパラメタ値を求める.
- (4) フラクタル形状 A の投影画像と正面図, 側面図の相互相関が十分大きければ処理を終了し, そうでなければ, (2) に戻る.

## 4. 5 3次元樹木形状の再構成方法の評価

本節では、樹木画像を入力とするフラクタルモデルの自動再構成方法の評価するため、まず、フラクタルモデルにより生成された樹木形状のシルエットを入力とした場合に、再構成されたフラクタル形状に対し、元のフラクタル形状のシルエットとの適合度などを評価した。つぎに、実写樹木画像から切り出したシルエットを用いて元の実際の樹木のフラクタルモデルを推定した結果を示す。

### 4. 5. 1 フラクタル形状を入力に用いた評価とその結果

ここでは、図4.11に示すように、モデルの縮小写像の個数が様々に異なるフラクタル形状から生成された入力画像に対し、モデルの縮小写像の個数を5個としてモデル推定を行うことで、妥当なフラクタル形状が生成されるかについて評価を行った。

- (1) 正面、側面図における、入力画像と推定されたフラクタル形状の適合度について。
- (2) 正面、側面以外の方向から見た場合の、元の3次元形状と推定されたフラクタル形状の適合度について。
- (3) 入力画像の生成に用いたフラクタル形状のフラクタル次元と、推定されたフラクタル形状のフラクタル次元の相関について。

上記の評価には、フラクタルモデルのパラメタをランダムに与えて形状を生成し、CGを用いて正面と側面への投影画像をつくり、それらを64×64画素のシルエット画像に変換して入力とした。フラクタルモデルについては、モデルの縮小写像の個数が、推定に用いるモデルと同じ5個の場合と、それと異なる4個、6個の場合で各17種類、全部で51種類を用いた。これらは、表4.1(a), (b), (c)に示す、シンメトリーなフラクタル形状を生成する縮小写像に対し、平均0、標準偏差16%の変動を与えることで生成した。入力画像の作成に用いた3次元形状の変動については、Box-Muller法 [117]により正規乱数を発生させることで得た。

入力画像をCGを用いて生成した理由は、画像の樹木部分と背景部分の分離が容易なこと、任意の入力樹木画像を生成できること、そして入力画像生成に用いた形状と、再構成形状の任意方向からの適合度の比較が容易であることによる。

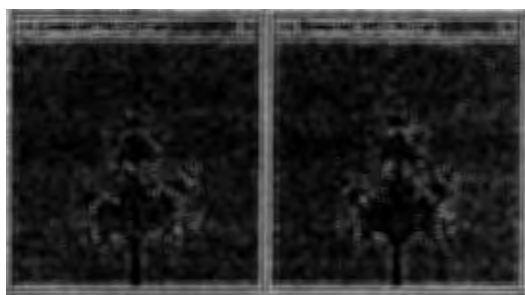
このようにして生成されたフラクタル形状から作られた入力画像を用いて、フラクタル形状を推定し、その際の入力画像とフラクタル形状の適合度をプロットした。



表 4. 1 フラクタルモデルの生成に用いた基本モデル

	sn	yn	$\theta$ zn	$\theta$ yn
(a)	0.50	0.40	0.00	0.00
	0.50	0.40	45.0	0.00
	0.50	0.40	45.0	120.0
	0.50	0.40	45.0	240.0
(b)	0.50	0.40	0.00	0.00
	0.50	0.40	45.0	0.00
	0.50	0.40	45.0	90.0
	0.50	0.40	45.0	180.0
	0.50	0.40	45.0	270.0
(c)	0.50	0.40	0.00	0.00
	0.50	0.40	45.0	0.00
	0.50	0.40	45.0	72.0
	0.50	0.40	45.0	144.0
	0.50	0.40	45.0	216.0
	0.50	0.40	45.0	288.0

図4.12のSample4.1, Sample4.2は, 上記評価に用いたフラクタル形状の入力画像例である. Sample4.1は, 本手法で樹木形状の近似に用いるのと同じ, 縮小写像5個よりなるフラクタルモデルであり, またSample4.2は, それとは異なり, 縮小写像6個よりなるフラクタルモデルである. 表4.2に, 各Sampleのフラクタルモデルのパラメタ値を示す.



Front

Side

Sample4.1



Front

Side

Sample4.2

図4.12 入力画像の例

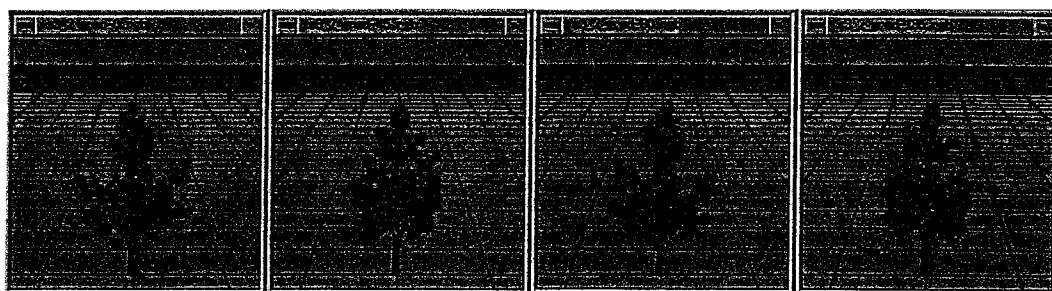
表 4. 2 入力画像例の生成に用いたフラクタルモデル

	sn	yn	$\theta$ zn	$\theta$ yn
Sample4.1	0.47	0.40	0.04	0.23
	0.49	0.39	44.21	3.66
	0.55	0.40	35.45	180.00
	0.53	0.38	35.3	81.48
	0.50	0.38	38.22	270.38
Sample4.2	0.45	0.34	23.36	316.06
	0.62	0.29	46.96	3.07
	0.48	0.30	107.14	69.53
	0.37	0.32	1.45	299.66
	0.49	0.32	44.02	233.65
	0.71	0.20	102.76	266.34

図 4. 1 3 の Sample4.1, Sample4.2 は, 推定されたフラクタルモデルにより 3 次元の樹木形状を生成し, 3 次元 CG 表示した結果である. また表 4. 3 に, 本手法により推定されたフラクタルモデルのパラメタの値を示す. 図 4. 1 3 では, 表 4. 3 に示すパラメタを用い, 3 章で示した樹木形状生成方法と同様に, 式 (4. 9) に従って樹木形状を生成した.

$$T = \bigcup_{m=1}^n \omega_m(T) \cup OS = \lim_{k \rightarrow \infty} \bigcup_{i=1}^k \bigcup_{(j_1, \dots, j_k) = (1, \dots, 1)}^{(n, \dots, n)} \omega_{j_1} \cdots \omega_{j_k}(OS) \cup OS \quad (4. 9)$$

式 (4. 9) の OS として六角錐台を用い, 縮小写像を OS に対し再帰的に適用することで生成される枝の長さが, 樹木の高さの 2% になったところで, 縮小写像の適用を終了し, その枝の末端に菱形の葉形状を付与している (図 4. 1 4). 図 4. 1 3 では, 樹木の正面方向, 側面方向からのほか, 45 度方向, 135 度方向からの 3 次元 CG 表示を行なった. 2 枚の樹木のシルエットから, 3 次元 CG 表示可能な形状が得られていることが分かる.



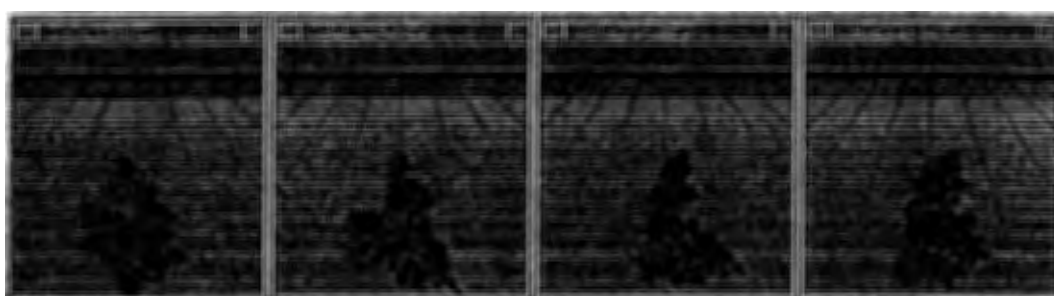
Front

45 (deg)

Side

135 (deg)

Sample4.1



Front

45 (deg)

Side

135 (deg)

Sample4.2

図 4 . 1 3 推定されたフラクタルモデルにより生成された樹木形状の 3 D C G 表示例

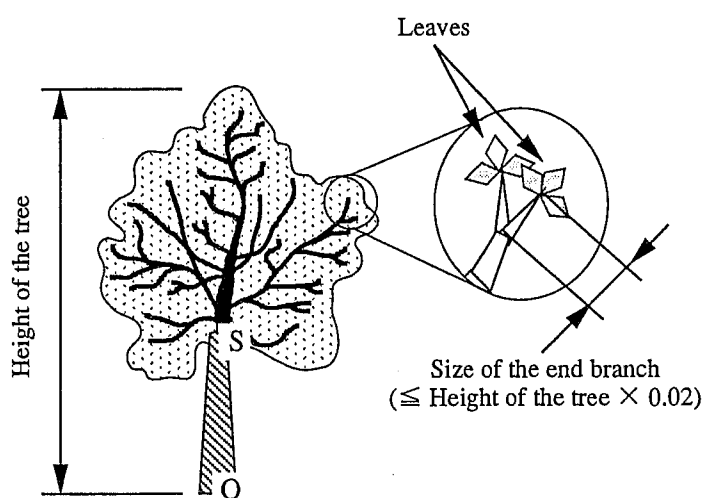


図 4 . 1 4 樹木形状の生成方法

表 4. 3 推定されたフラクタルモデル

	sn	yn	$\theta$ zn	$\theta$ yn
Sample4.1	0.51	0.36	1.94	303.18
	0.50	0.42	41.01	1.19
	0.53	0.47	37.67	182.60
	0.53	0.40	31.71	91.22
	0.47	0.44	36.31	272.47
Sample4.2	0.63	0.45	20.82	323.74
	0.63	0.31	44.73	6.31
	0.69	0.00	19.07	199.51
	0.39	0.69	114.66	103.82
	0.70	0.43	97.26	282.12

また表 4. 4 (1) に、入力に用いた正面と側面のシルエットと、再構成されたモデルで生成された形状を投影して得られたシルエットの、正規化された相互相関の正面と側面での値の平均を示す。表 4. 4 (1) から、入力に用いたシルエットの元の形状が、推定に用いたのと同じ、5 個の縮小写像よりなるフラクタルモデルにより生成された Sample4.1 と、6 個の縮小写像よりなるフラクタルモデルにより生成された Sample4.2 の場合で、同じような適合度で、フラクタルモデルが推定されていることが分かる。

表 4. 4 (2) は、入力画像の生成に用いた形状と、再構成されたモデルで生成された形状の、45 度方向、135 度方向に投影して得られるシルエット画像の、正規化された相互相関の値を示す。入力は正面図と側面図であったが、45 度方向、135 度方向においても同じような適合度を与えるフラクタルモデルが推定されていることが分かる。



表 4. 4 入力画像と生成された樹木形状の投影画像との間の相互相関

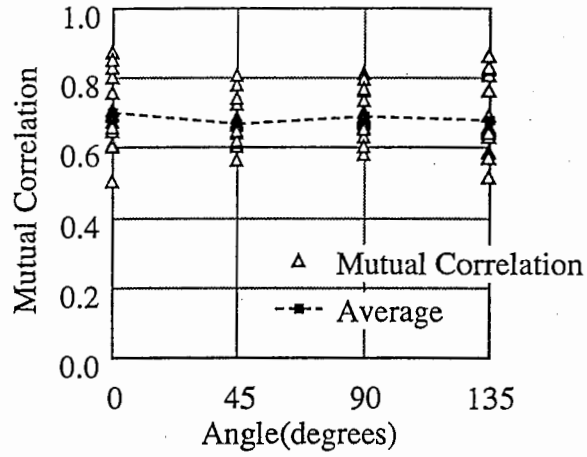
(1) 正面図と側面図での比較

	Sample4.1	Sample4.2
Front View	0.81	0.85
Side View	0.82	0.84

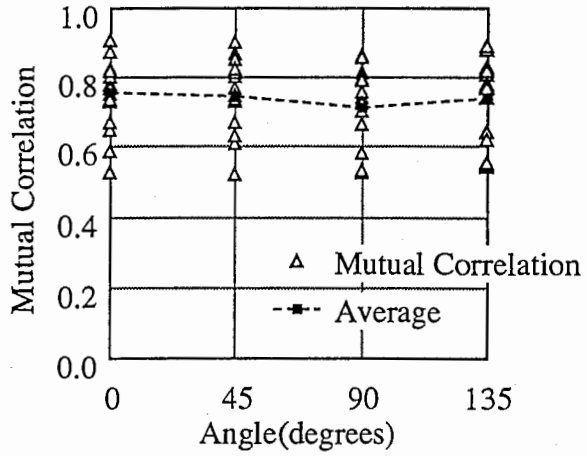
(2) 他の方向から見たときの投影画像上での比較

	Sample4.1	Sample4.2
45(deg)	0.86	0.85
135(deg)	0.85	0.84

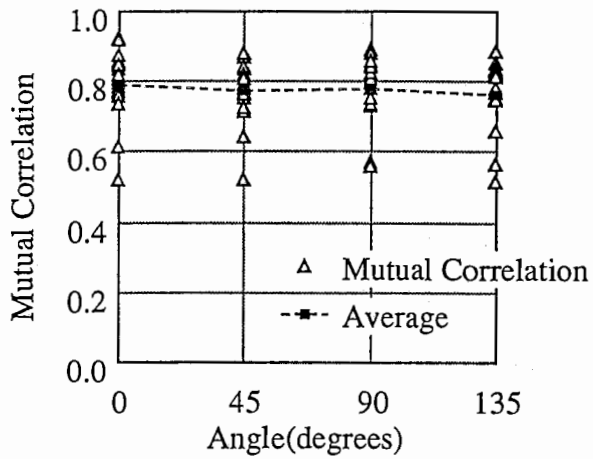
図 4. 15 (a) ~ (c) は、評価に用いた 51 種類の入力画像に対し、推定されたフラクタルモデルにより生成された形状を、正面 (0 度方向)、側面 (90 度方向)、そして斜め方向 (45 度方向、135 度方向) に投影して得られたシルエット画像と、元の 3 次元形状のシルエット画像の間の相互相関の計算結果である。(a) は、入力画像の作成に用いた 3 次元形状のモデルの縮小写像の個数が 4 個の場合、(b) は 5 個の場合、(c) は 6 個の場合の結果である。また、横軸は形状の投影方向、縦軸は相互相関値を示す。グラフ中の点線は各方向での相互相関の平均値を結んだものである。



(a) Four Contraction Mappings in IFS of Original Shape



(b) Five Contraction Mappings in IFS of Original Shape



(c) Six Contraction Mappings in IFS of Original Shape

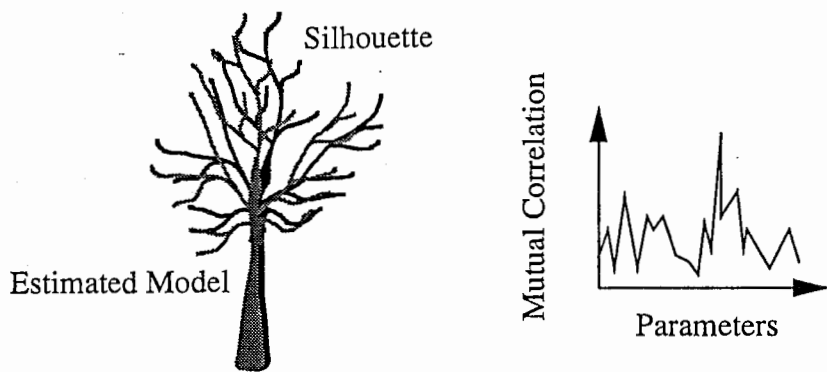
図 4. 15 正面, 側面, 及び他の方向から投影した場合の相互相関

(1) 入力画像と推定されたフラクタル形状との相互相関について

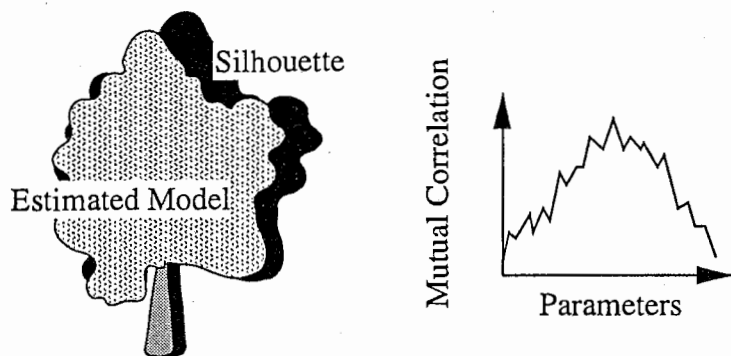
図4.15 (a) ~ (c) において、フラクタル形状の推定に用いた正面 (0度方向)、側面 (90度方向) の相互相関の平均値は、(a) で0.697、(b) で0.733、(c) で0.787であり、元の3次元形状の縮小写像の個数が少ない(a) の場合に低い値となった。モデルの縮小写像の個数が少ないということは、一般的に、生成される形状のフラクタル次元が低いということの意味する。本手法で用いた樹木形状モデルでは、葉と枝がまばらな樹木形状となる。そのような形状のシルエットにモデルを合わせ込む場合、シルエットとして露呈している枝に正確に適合しない限り、高い相互相関は得られない(図4.16 (a))。しかし計算誤差などの要因から、最急降下法などにより真のピークに到達することが難しい。結果として推定の精度が落ちたと考えられる。一方、図4.16 (b) の場合には、真のピークに到達できなくても、比較的安定して高い相互相関が得られると考えられる。

(2) 正面、側面から見た場合以外の適合度について

図4.15 (a) ~ (c) から、フラクタル形状の推定に用いた正面 (0度方向)、側面 (90度方向) での相互相関の値と、それ以外の方向 (45度方向、135度方向) で、ほとんど差がないことが分かる。このことから、本手法により推定されるフラクタルモデルが生成する形状は、推定に用いる正面図、側面図と適合するだけでなく、他の方向から見た場合でも、入力画像を生成するのに用いた形状によく一致していると言える。



(a) Silhouettes of Tree with little leaves



(b) Silhouettes of Leafy Tree

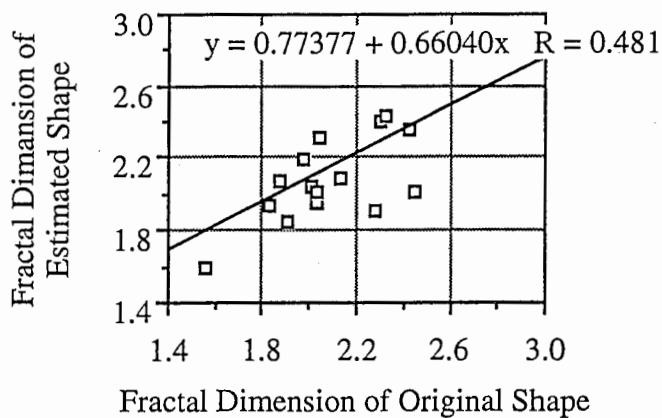
図 4. 1 6 モデルのパラメタの変更が相互相関に与える影響

つぎに、元の3次元形状と、推定されたフラクタル形状のフラクタル次元を比較した結果を、図4.17に示す。入力画像は、(1)と(2)の評価で使用したものと同一である。図4.17(a)は、入力画像の作成に用いた3次元形状のモデルの縮小写像の個数が4個の場合、(b)は5個の場合、(c)は6個の場合の結果である。横軸は入力画像を生成するのに用いた3次元形状のフラクタル次元、縦軸は推定されたフラクタル形状のフラクタル次元を示す。グラフ中の直線は、実験結果を直線回帰した結果である。Rは標本相関係数を表す。

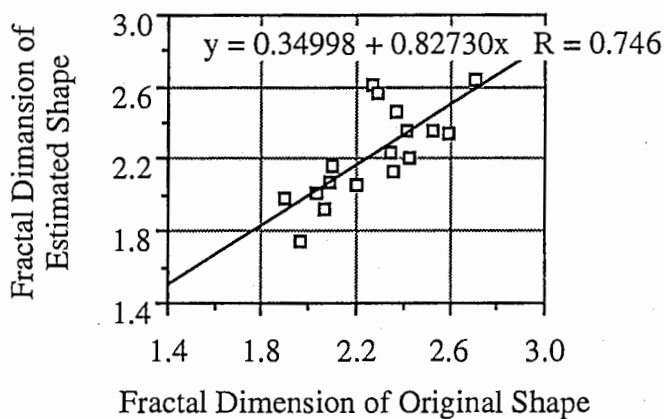
### (3) 元形状のフラクタル次元と推定された形状のフラクタル次元の比較

(b)、(c)では、Rはそれぞれ0.746、0.724と比較的高い値であり、入力シルエットの生成に用いたフラクタル形状と同程度のフラクタル次元をもつフラクタル形状が推定されていることを表している。一方(a)では、Rは0.481であり、入力シルエットの生成に用いたフラクタル形状のフラクタル次元と、推定されたフラクタル形状のフラクタル次元の相関は低い。これは(1)において、(a)の場合に、入力画像と推定されたフラクタル形状の間の相互相関が低くなったのと同様の理由で、形状の推定の精度が悪くなったからと考えられる。

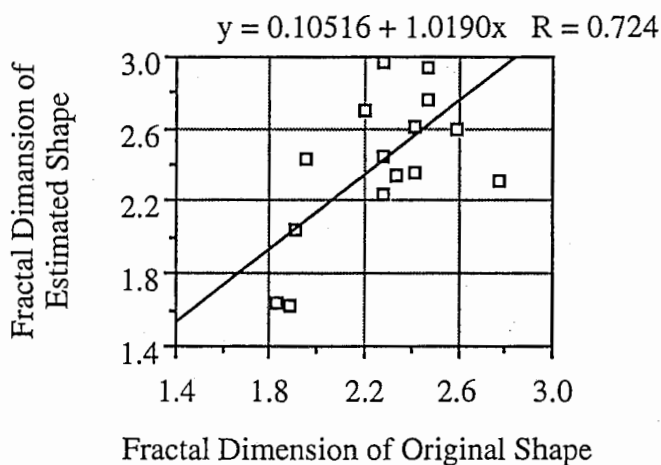
このように、3次元樹木形状をフラクタルでモデル化し、樹木形状の正面図と側面図に適合する形状を生成するよう、モデルのパラメタを推定することで、従来のシルエットを用いた3次元形状再構成法では扱えなかった、シルエットの内部に複雑な構造を持つ樹木形状に対して、意図した3次元樹木形状が生成できることが分かる。



(a) Four Contraction Mappings in IFS of Original Shape



(b) Five Contraction Mappings in IFS of Original Shape



(c) Six Contraction Mappings in IFS of Original Shape

図 4. 17 元の形状と推定された形状のフラクタル次元の比較

#### 4. 5. 2 実写樹木画像を入力とした場合の推定結果

本節では、実写樹木画像から作成したシルエット画像を入力とした場合の、推定されたフラクタル形状の3DCG表示例を示す。

図4.18(a)～(d)に、実写樹木画像から切り出された樹木のシルエット画像と、それを入力として推定されたフラクタル形状を示す。表4.5は、図4.18(a)～(d)で推定された形状のフラクタル次元と、それぞれのシルエットと推定された形状の相互相関の値、そしてシルエット部分の画素の割合を示したものである。(b)～(d)については、推定された形状のフラクタル次元が大きいほど、シルエットと推定された形状の相互相関の値は高く、4.5.1での、元形状にフラクタル形状を用いた場合の形状推定と同様の傾向が得られている。また、各例について、実写樹木のシルエット画像のもつ雰囲気、推定されたフラクタル形状でよく再現されていることが分かる。(a)については、推定された形状のフラクタル次元は、(b)、(c)と同程度であるが、(b)、(c)に比べて、葉の少ないシルエット画像であることから、シルエット画像と推定形状の相互相関が低くなったと考えられる。

表4.6に、本手法により推定されたフラクタルモデルのパラメタの値を示す。3DCG表示に使用した樹木形状は、表4.6に示すパラメタを用い、式(4.9)に従って樹木形状を生成した。4.5.1での樹木形状の3DCG表示例の場合と同様に、式(4.9)のOSとして六角錐台を用い、縮小写像をOSに対し再帰的に適用することで生成される枝の長さが、樹木の高さの2%になったところで、縮小写像の適用を終了し、その枝の末端に菱形の葉形状を付与している。

(a)



(b)



(c)



(d)



Actual Tree Image

Input Silhouettes

CG Images of Fractal Shape estimated

図 4. 1 8 実写樹木画像のシルエット画像を入力とした場合の推定結果  
(3DCG表示例)



表 4. 5 実写樹木画像のシルエット画像を入力とした場合の推定結果 (相互相関)

	Fractal Dimension	Mutual Correlation		Silhouettes Region $g_i$	
		Front	Side	Front	Side
(a)	2.371	0.622	0.643	0.170	0.140
(b)	2.308	0.740	0.791	0.258	0.227
(c)	2.334	0.749	0.846	0.454	0.438
(d)	2.913	0.864	0.819	0.357	0.319

表 4. 6 実写樹木画像のシルエット画像から推定されたフラクタルモデル

	sn	yn	$\theta$ zn	$\theta$ yn
(a)	0.521	0.430	4.374	331.946
	0.395	0.335	35.369	7.901
	0.606	0.289	52.536	181.791
	0.493	0.284	28.359	105.854
	0.523	0.355	42.903	273.093
(b)	0.593	0.409	26.477	158.883
	0.361	0.318	48.686	355.916
	0.490	0.300	39.659	176.279
	0.604	0.241	39.657	83.015
	0.328	0.393	78.280	267.346
(c)	0.460	0.554	23.289	304.851
	0.650	0.096	28.134	28.089
	0.650	0.165	31.410	203.145
	0.486	0.554	76.907	107.099
	0.557	0.320	49.480	286.110
(d)	0.675	0.314	5.632	252.316
	0.474	0.191	46.662	357.374
	0.650	0.063	32.704	181.364
	0.650	0.132	33.827	95.482
	0.460	0.223	38.276	280.367

## 4. 6 まとめ

本章では、3次元樹木形状の正面図と側面図画像を入力として、それによく適合する3次元樹木形状を生成するフラクタルモデルの、縮小写像のパラメタを自動的に推定する方法を提案した。具体的には、3次元樹木形状の正面と側面のシルエットに対し、フラクタルモデルによって生成されるフラクタル形状を投影して相互相関を計算し、それを最大する縮小写像のパラメタを推定する。

このような推定を可能にするため、各シルエットに割り当てる縮小写像を3個に制限することを提案した。また、推定には最適傾斜法を用いたが、最適傾斜法の結果は初期値に依存するので、適当な初期値を得るための方法を提案した。

つぎに、この3次元樹木形状のシルエット画像を入力とするフラクタル形状の自動推定方法を評価するために、様々なフラクタル次元を持つフラクタル樹木形状の正面図、側面図を入力として、フラクタル形状の自動推定を行なった。さらに、本手法の拠り所であるコラージュ定理は、フラクタル形状でないものを、フラクタル形状で近似できることを示していることから、実写景観画像中の樹木に対しても、本手法が有効に機能すると考え、実写樹木画像を入力として、その3次元形状の推定を行なった。

### (1) フラクタル形状を用いて作成したシルエット画像入力とした場合

入力シルエット画像を作成するのに使用した形状の、フラクタルモデルの縮小写像の個数が少ない4個の場合に、入力シルエット画像と推定された形状の相互相関が低くなる結果が得られた。しかし、入力シルエット画像と推定された形状の相互相関が比較的高い場合には、正面、側面以外の方向のシルエット画像に対しても、正面、側面のシルエット画像に対するのと同程度の相互相関が得られた。

さらに、シルエット画像を作成するのに用いた形状と、推定された形状のフラクタル次元の相関が高いという結果から、本手法で推定されたフラクタル形状は、シルエット画像を作成するのに用いた元の形状を3次元的によく近似していると考えられる。図4.12に示したSample4.1、Sample4.2の入力画像と、図4.13の再構成結果から、形状の再構成が良好に行なわれたことがわかり、上記の結果を裏付けている。

### (2) 実写画像から作成したシルエット画像入力とした場合

フラクタル形状を用いて作成したシルエットを入力画像としたときと同様、入力画像に似たシルエットを持つフラクタル形状が推定できた。

以上, 従来のシルエットを用いた3次元形状再構成法では扱えなかった, シルエットの内部に複雑な構造を持つ樹木形状に対して, 意図した3次元樹木形状が生成できることを示した.

## 5. 視覚特性を用いた樹木画像生成方法とその評価 [56] [58]

### 5. 1 まえがき

本章では、フラクタルを用いて簡略化された樹木形状を用いて、樹木の多数存在する仮想空間を3DCGにより表示する際に、観察者に簡略化形状を用いることによる画質の劣化を意識させずに、3次元画像を生成する方法について述べる。

まず最初に、フラクタルを用いて簡略化した形状に対し、樹木画像をテクスチャマップすることで、形状を大きく簡略化しても、利用者にそれを意識させない表示が可能であることを、主観評価実験により明らかにする。そして、この考え方による、簡略化形状用のテクスチャデータを用いたレンダリング方法を提案する。

また、実時間CG表示が可能な程度にシーン全体の樹木形状データを削減するためには、シーン中の樹木それぞれに与える形状粗さを制御する方法を検討する必要がある。そこで、樹木の多数存在する仮想空間を表示する場合に、利用者の視点位置などを検出し、それをもとに中心視/周辺視などの視覚特性を利用して、表示面上で重要な部分のみを高品質に表示することで、表示に必要な形状データ量が大幅に減少することを示す。

つぎに、(1)フラクタルを用いた樹木形状の簡略化手法、(2)テクスチャマップを用いた3次元樹木画像の生成方法、そして、(3)中心視、周辺視の性質を利用した形状粗さの制御方法を組み合わせた仮想空間表示評価システムを実装し、その効果を確認したので、その結果について述べる。

### 5. 2 テクスチャマップを用いた簡略化形状の表示方法

本節では、まず、フラクタルを用いて簡略化した形状に対し、樹木画像をテクスチャマップすることで、形状を大きく簡略化しても、利用者にそれを意識させない表示が可能であることを、主観評価実験により明らかにする。つぎに、この考え方による、簡略化形状用のテクスチャデータを用いたレンダリング方法を提案し、樹木の風による揺らぎなど、形状の変形を伴う表示の際にも違和感のない表示が可能になることを示す。

#### 5. 2. 1 テクスチャマップを用いた簡略化形状の表示

ここでは、簡略化形状に対して、2次元樹木画像をテクスチャマップして表示する方法の概要を

示す(図5.1). 形状粗さを大きく指定した粗い形状データ(図5.1(b))に対して, 細部の形状や色調を与えるために, 詳細な形状データ(図5.1(a))を用いて生成した2次元樹木画像をテクスチャマップする.

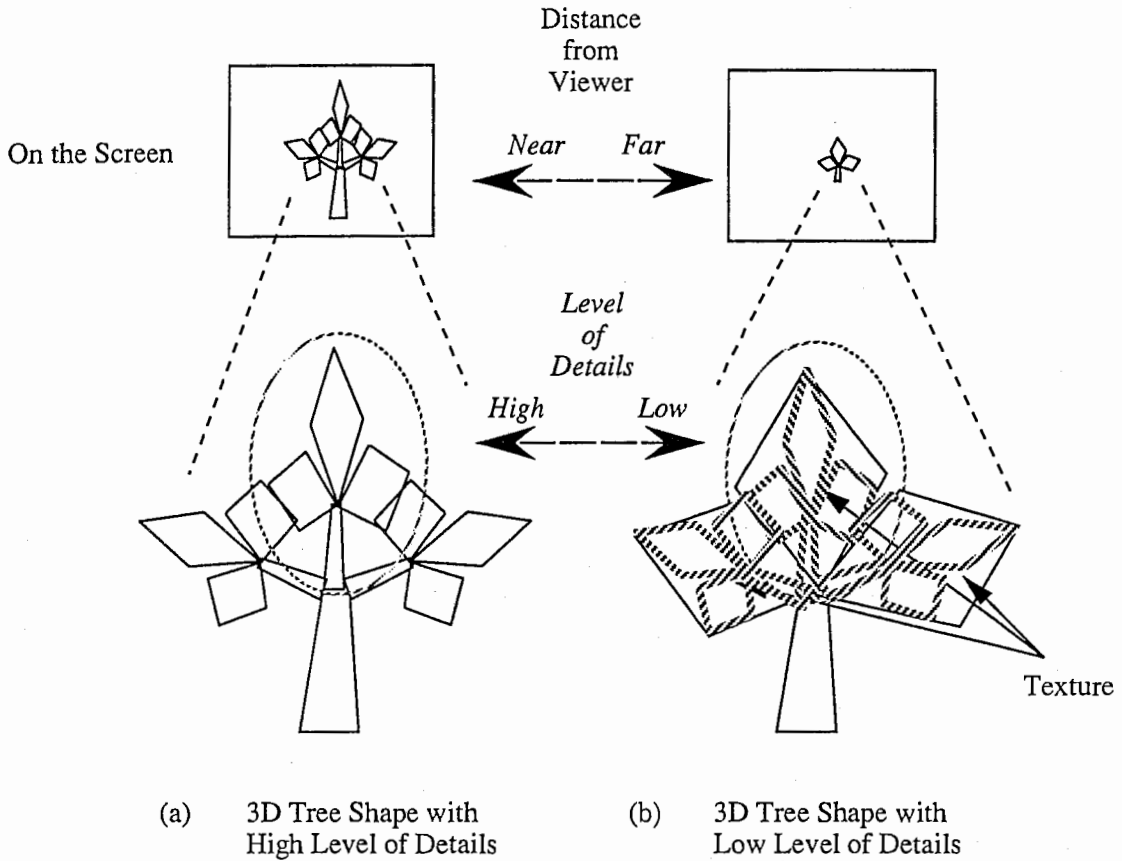


図5.1 テクスチャマップを用いた簡略化形状の表示

図5.2に示すように, まず2次元樹木画像は, テクスチャ空間にマッピングされる(式(5.1)). 式(5.1)において,  $(x_{image}, y_{image})^T$ は画像中での画素の位置,  $I_x, I_y$ は画像の横と縦の画素数,  $(i_{tex}, j_{tex})^T$ はテクスチャ空間中での画素の位置を表す.

$$\begin{pmatrix} i_{tex} \\ j_{tex} \end{pmatrix} = \begin{pmatrix} 1/I_x & 0 \\ 0 & 1/I_y \end{pmatrix} \begin{pmatrix} x_{image} \\ y_{image} \end{pmatrix} \quad (5.1)$$

また, 樹木形状の葉および簡略化された形状も同様にテクスチャ空間にマッピングする(式(5.2)). 式(5.2)において,  $(x_{shape}, y_{shape}, z_{shape})^T$ はモデル空間中での形状データの座標値,  $S_x, S_y$ はモデル空間とテクスチャ空間のスケール比,  $(s_{xtex}, s_{ytex})$ はテクスチャ空間中での形状データの座標値を表す.

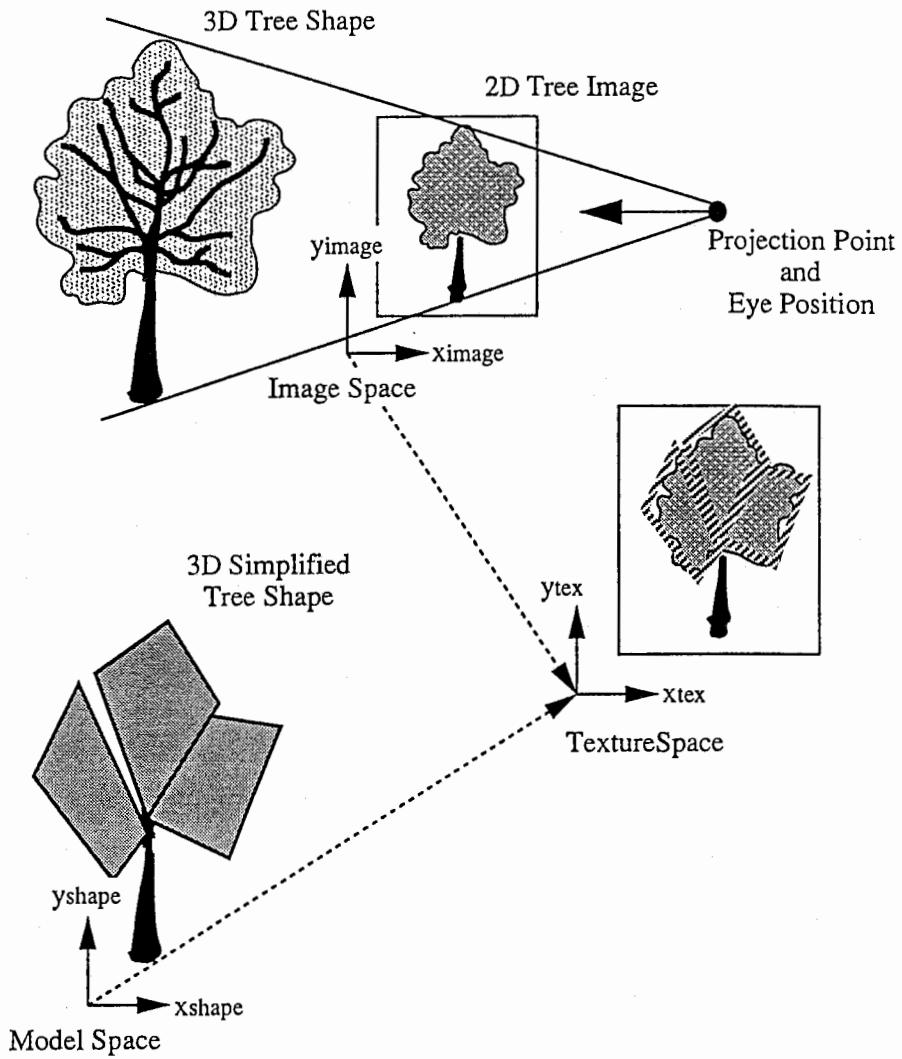
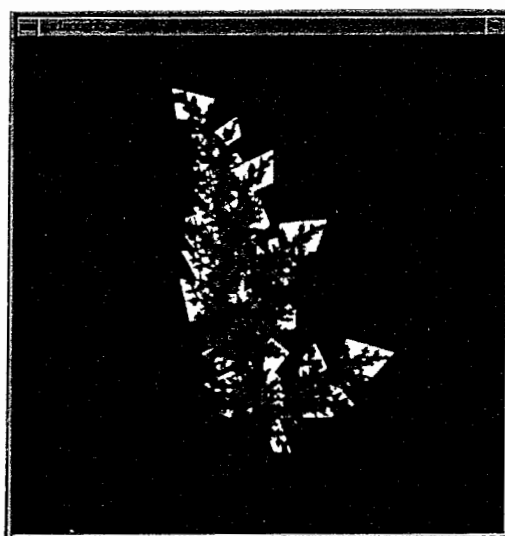
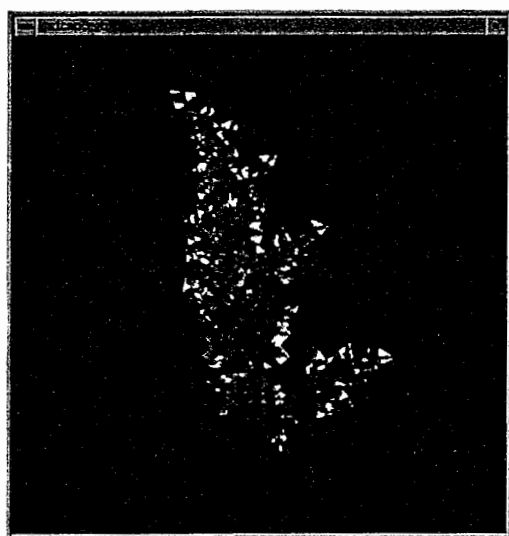


図 5. 2 2次元樹木画像の簡略化形状へのテクスチャマップ

$$\begin{pmatrix} Sx_{tex} \\ Sy_{tex} \end{pmatrix} = \begin{pmatrix} 1/Sx & 0 & 0 \\ 0 & 1/Sy & 0 \end{pmatrix} \begin{pmatrix} x_{shape} \\ y_{shape} \\ z_{shape} \end{pmatrix} \quad (5.2)$$

上記のようにして2次元樹木画像との対応付けを行なったのち、汎用GWS(グラフィックワークステーション)の機能を用い、形状データに画像データをテクスチャマップして表示する。

図 5. 3 は、3章で用いた樹木形状である Sample3.1 について、樹木画像のテクスチャマップの様子を示すため、樹木画像の背景部分(灰色)を不透明、かつ3次元CG表示の背景色を黒として表示した例である。図 5. 3 (1) は形状粗さ  $Q = 0.04$ 、図 5. 3 (2) は形状粗さ  $Q = 0.08$  の3次元形状に対してテクスチャマップを行なった。



(1) 3D Tree Shape (Q = 0.04)

(2) 3D Tree Shape (Q = 0.08)

図 5. 3 2次元樹木画像の簡略化形状へのテクスチャマップの表示例

## 5. 2. 2 テクスチャマップを用いた簡略化形状の表示方法の評価

ここでは、テクスチャマップを用いた簡略化形状の表示方法の評価として、樹木の表示サイズを変えることで、各表示サイズに対して指定できる形状粗さの上限を評価する実験を行ったので、これについて述べる。表示画像の品質として、2眼式立体視した場合の形状の立体感の度合を評価した。以下に実験条件を示す。

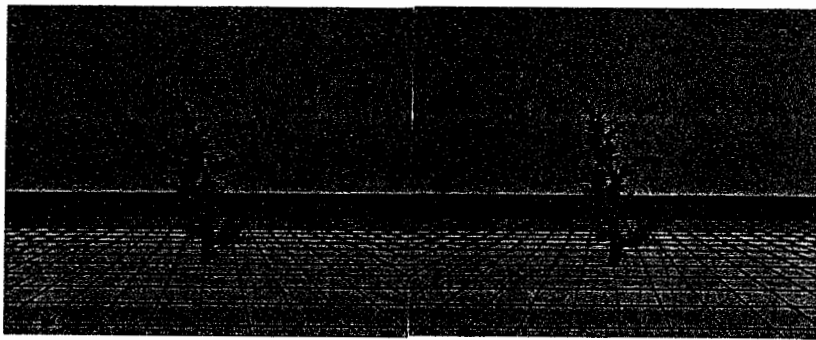
視点はスクリーンから50cmのところ、19inchスクリーン上に2眼式立体表示された画像を、液晶シャッター眼鏡を通して立体視した。左右画像の切り替えは120Hzで行った。実験には、3章で画像生成時間の評価で用いた、Sample3.1, Sample3.2の2種類の樹木形状を使用した。樹木表示位置は立体視の際の輻輳角を約 $0.16^\circ$ とし、視点からの距離250cmとした。樹木の表示サイズは画角を $25^\circ$ から $5^\circ$ の間を $5^\circ$ 刻みに変化させた。樹木の形状粗さは0.04から0.2の間を0.04刻みで変化させ、表示画像を2眼式立体視したときの立体感の度合を評価した。評価は、形状粗さを0.02として生成した形状と、指定する粗さで生成した形状を一对比較し、その違いを表5.1に示



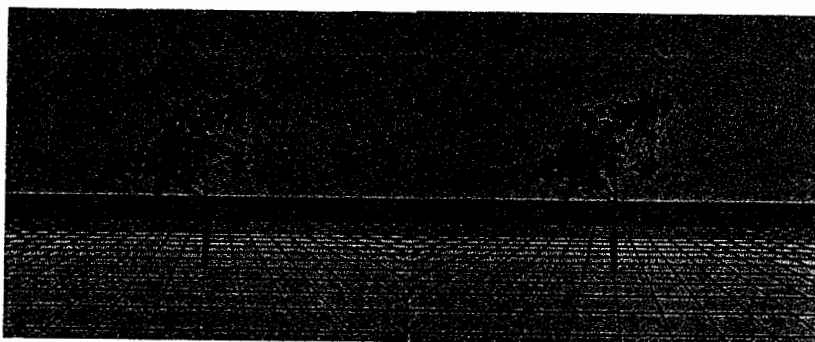
す5段階の評価値で主観評価した。被験者は10名で行った。本実験は、立体感の要因のうち、両眼視差のみを対象としたため、実験に使用した3次元CG表示は照明無しの場合で行なった。実験に用いた両眼視差画像例を図5.4に示す。樹木画像の表示に用いた計算機は、3章と同様に、Silicon Graphics社のCrimson REを用いた。

表5.1 主観評価に用いた評価値

5	元の画像に対して全く差を感じない
4	元の画像に対してほとんど差を感じない
3	元の画像に対して差を感じるが違和感はない
2	画像に違和感がある
1	画像に非常に違和感がある



Sample3.1



Sample3.2

図5.4 評価実験に用いた2眼式立体表示例(交差法)

図5.5は、上記の実験結果から得られた、各表示サイズにおいての観察者の許容できる形状粗さを示したものである。許容できる形状粗さとしては、評点3を下回らないこととし、95%の信頼区間の下限が3より大きい形状粗さの上限をプロットしてある。よって、いずれのサンプルにおいても、許容できる形状粗さQ以下では、十分な画質が得られるといえる。図5.5から、例えば両サンプルを表示サイズ15°以下で表示したとき、共に0.10程度の形状粗さで十分な画質が得られることが分かり、3章での形状粗さに対する簡略化形状のデータ量についての評価結果を考慮すれば、データ量を大幅に減少させられることが分かる。

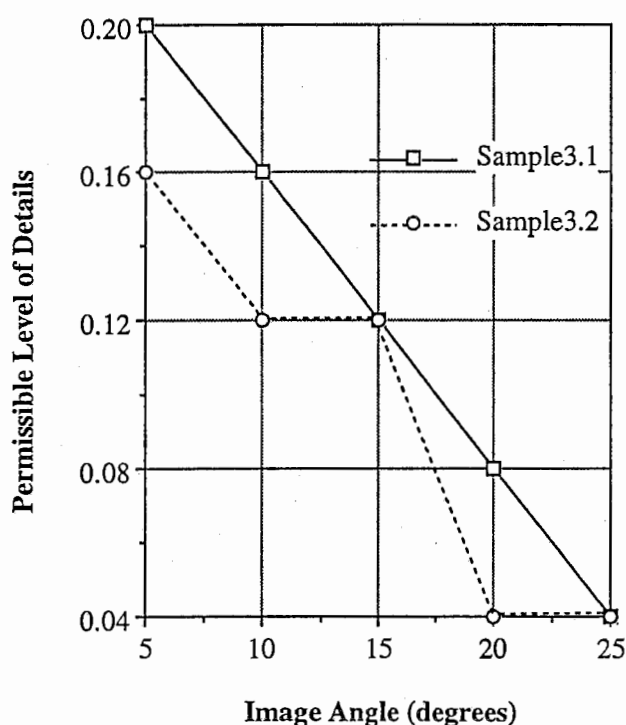


図5.5 樹木形状の表示画角に対する許容できる形状粗さ

### 5.2.3 簡略化形状用テクスチャを用いたレンダリング方法

上記のテクスチャマップを用いた簡略化形状の表示方法では、仮想空間中の視線方向とテクスチャマップに用いる2次元樹木画像の投影方向が比較的近い場合には、問題なく表示が行えるが、そうでない場合には、図5.6に示すように違和感のある表示画像になる。また、この方法は、樹木形状の入り組んだ部分のテクスチャ情報が無いために、観察者の視点移動、あるいは風による揺らぎの表示といったような形状データの変形により、入り組んだ部分が露呈するような状況において不具合が生じる。



図 5. 6 簡略化形状を用いて表示された樹木周りの視点移動例

そこで、図 5. 7 に示すように、フラクタルを用いて簡略化された樹木形状の各簡略化形状に対し、個別にテクスチャデータを用意しておくことにする。個別のテクスチャデータ（以後はテクスチャエレメントと呼ぶ）は、図 5. 8 に示すように、簡略化形状  $P$  に対して、 $P$  に対応する詳細な 3 次元形状を、 $P$  に対して平行投影することで得られる画像データのことである。簡略化形状の表示の際には、簡略化形状用テクスチャのテクスチャエレメント  $I$  を、 $P$  にテクスチャマップする。 $P$  は十字に組んだポリゴンであるが、表示の際には十字に組んだポリゴンのうち、その法線ベクトルと視線方向ベクトルのなす角が小さいほうを選択される。また、テクスチャデータとしては、簡略化形状の前後左右に対応する 4 種類を用意することにより、入り組んだ部分の細部の形状や色調を与える。

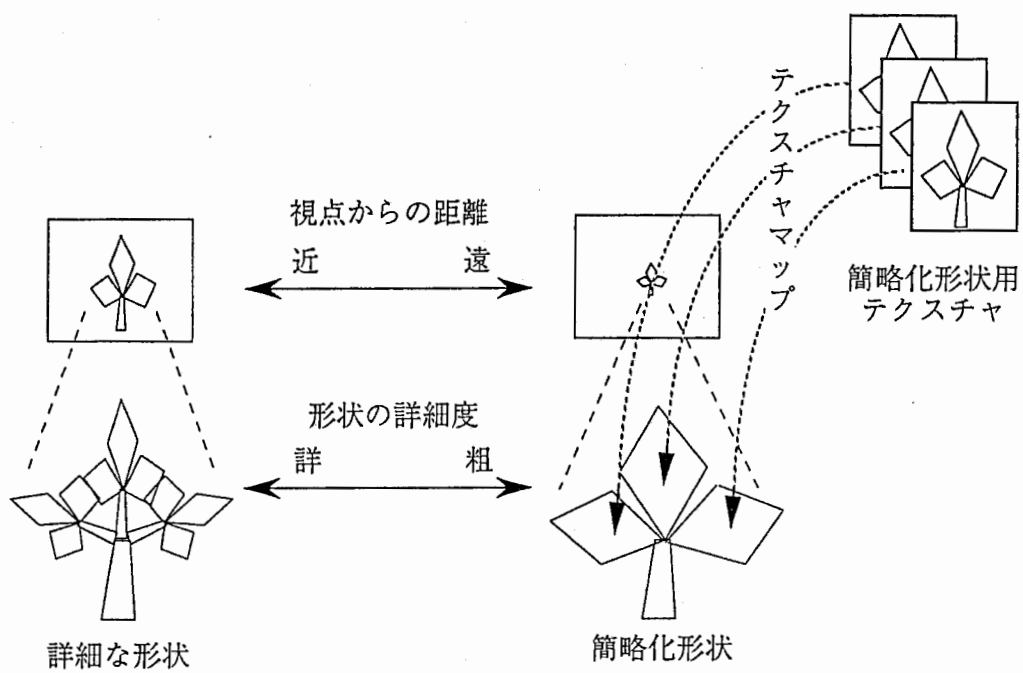


図 5. 7 簡略化形状用のテクスチャを用いた簡略化形状の表示

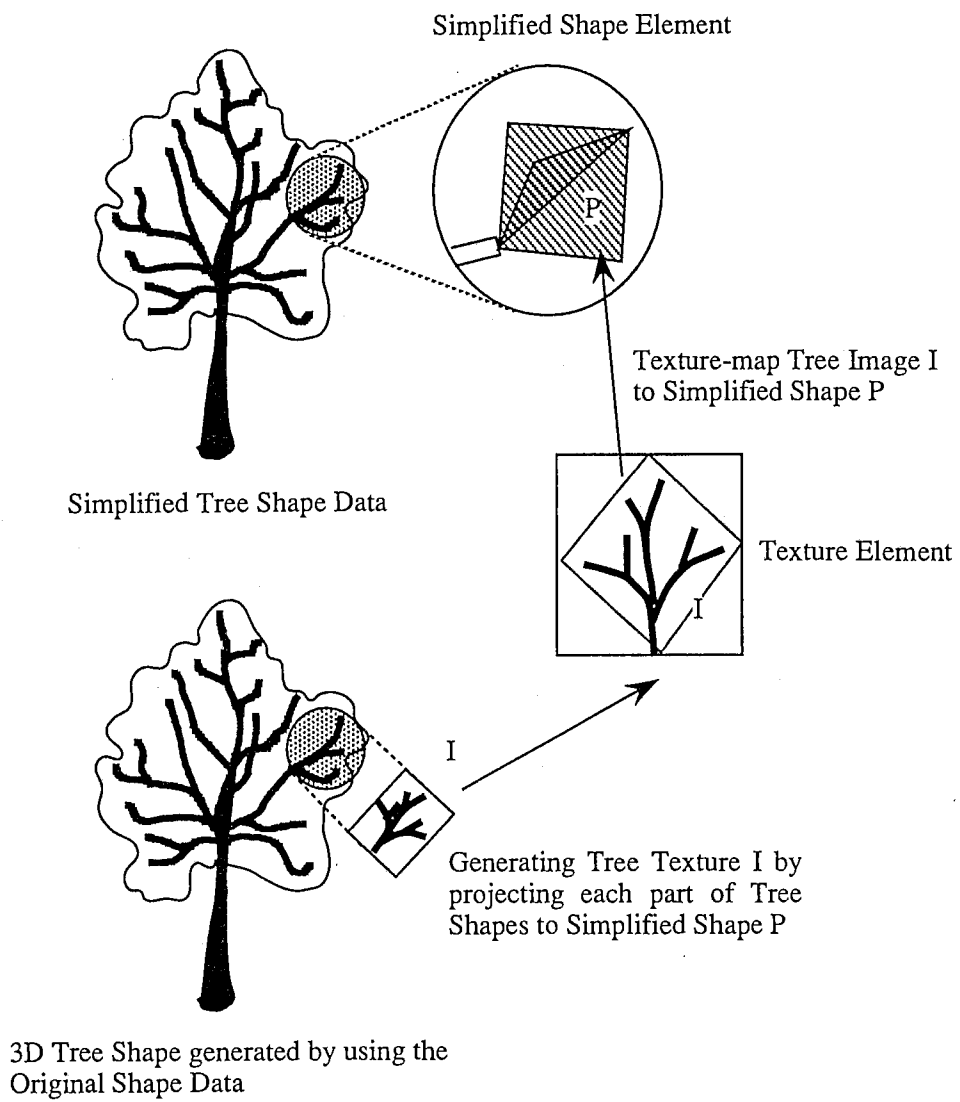


図 5. 8 簡略化形状用テクスチャデータの作成方法

#### 5. 2. 4 簡略化形状用テクスチャを用いた樹木画像生成例

図5.9(a)は、樹木の風による揺らぎのを、簡略化形状用のテクスチャデータを用いて画像生成したときの表示結果である。一方、図5.9(b)は、適当な視点位置から投影した2次元樹木画像を用いて、簡略化形状をレンダリングしたときの表示例である。図5.9では、(a)と(b)の差異を強調するため、(b)で使用した画像には、枝の部分を茶色、葉の部分を緑の2色で表示したものを用いた。また、揺らぎによる変形を分かりやすく表示するために、シンメトリーな樹木形状を表示に用いた。

図5.9(a)、(b)共に、揺らぎによる変形の結果、変形のない状態では見えなかった部分が露呈する。簡略化形状用のテクスチャデータを用いた図5.9(a)の場合には、そのような場合でも違和感のない表示が行なわれている。しかし、単純に2次元樹木画像を用いただけのレンダリングによる表示(図5.9(b))では不適切な表示結果となっている。具体的には、図5.9(b)の各フレームで丸で囲った部分の茶色い菱形のポリゴンは、樹木の詳細な形状を置き換えたポリゴンであり、本来は詳細な形状を投影して得られたテクスチャが貼られるべきであるが、2次元樹木画像は変形のない状態で見えない部分についての情報を有しないため、茶色一色で塗り潰されている。



(a) 簡略化形状用テクスチャをテクスチャマップした表示



(b) 2次元樹木画像をテクスチャマップした表示

図 5. 9 簡略化形状用のテクスチャデータを用いた揺らぎの表示

### 5. 3 視覚特性を用いた形状粗さの制御方法

フラクタルを用いた形状簡略化, 及びテクスチャマップを用いた樹木画像生成により, 樹木の多数存在する仮想空間の表示を行う場合には, 実時間CG表示が可能な程度にシーン全体の樹木形状データを削減するよう, シーン中の樹木それぞれに与える形状粗さを制御する方法を検討する必要がある. 例えば, 形状粗さを以下のように制御することが考えられる.

- (1) 表示面上で一様の表示品質が得られるよう, それぞれの樹木の形状粗さを制御する.
- (2) 中心視, 周辺視, あるいは動態視力などの視覚特性を用いて, 表示面上で重要な部分のみを高品質に表示するよう, それぞれの樹木の形状粗さを制御する.

インタラクティブな景観シミュレーションを対象としたリアルタイムCGの分野では, 上記(1)の考え方により, 表示面上での形状の最小のサイズが一定となるよう, シーン中の形状の形状粗さを制御する方法が検討されているが, 仮想空間表示の場合には, 利用者の視点位置などを検出し, それをもとに視覚特性を利用する, 上記(2)の制御方法が有望であると考えられている [36]. すなわち, 人間の視覚には, 網膜上の視細胞の分布状態により, 注視している方向付近は詳細に識別できるが(中心視), 視野の周辺部分では分解能が指数関数的に低下する(周辺視)ことが知られている [85-88]. この視覚特性を利用した表示装置の研究として, 実写静止画像の広視野かつ高精細な表示を実現するために, 視線検出を行ない, 分解能の高い注視点付近に高精細画像を表示し, それ以外の周辺部分には解像度の粗い画像を重ね合わせる手法が報告されている [118]. そこで, リアルタイムCGの分野でも, 中心視, 周辺視, 動態視力, そして融像領域を考慮して, 視力の低下に応じて3次元CG表示する物体の形状を簡略化する手法が提案されている [36].

この手法では, 階層化適応型三角形パッチ生成法 [89] を使用して, 最も精細な形状データから順次精細度を落とした形状データを事前に用意し, 表示の際の精細度は, 視力の平方根に形状のポリゴン数が比例するよう制御している. しかし, 視覚特性のモデル, あるいは視力に応じた形状の精細度の制御方法の妥当性については, 今後の検討課題とされている. また, これを一般的なシーンに適用した場合のデータ量の削減効果についての評価は行なわれていない.

そこで, 本節では, フラクタルを用いた高速3次元樹木画像生成法により, 仮想空間中に樹木が一様に分布しているシーンを表示する場合に, 樹木形状に与える形状粗さを, 上記(1), (2)それぞれの方法で制御したときの, 形状データ量をシミュレーションによって評価した. その結果, フラクタルを用いた樹木形状の簡略化による表示においては, 上記(2)のように視覚特性, 特に



中心視と周辺視を用いて形状粗さを制御することで、シーン全体の樹木形状データ量を大幅に削減できることを示す。

### 5. 3. 1 視点からの距離に応じた形状粗さの制御

前述の、樹木の表示サイズに対する許容できる形状粗さについての主観評価実験では、実験に使用した2つのサンプルに対して、視点より遠方に表示され、表示サイズが小さいほど、大きな形状粗さを指定できることを示している。

樹木の仮想空間中でのサイズ $L_0$ と、表示面上でのサイズ $S_{tree}$ 、そして視点から樹木までの距離 $D$ は、式(5.3)と書けることから、図5.10に、図5.5の横軸を $D/L_0$ として、許容できる形状粗さを視点からの距離(ただし樹木サイズで正規化)の関係として示す。

$$\frac{D}{L_0} = \frac{1}{\tan(S_{tree})} \approx \frac{180}{\pi \cdot S_{tree}} \quad (5.3)$$

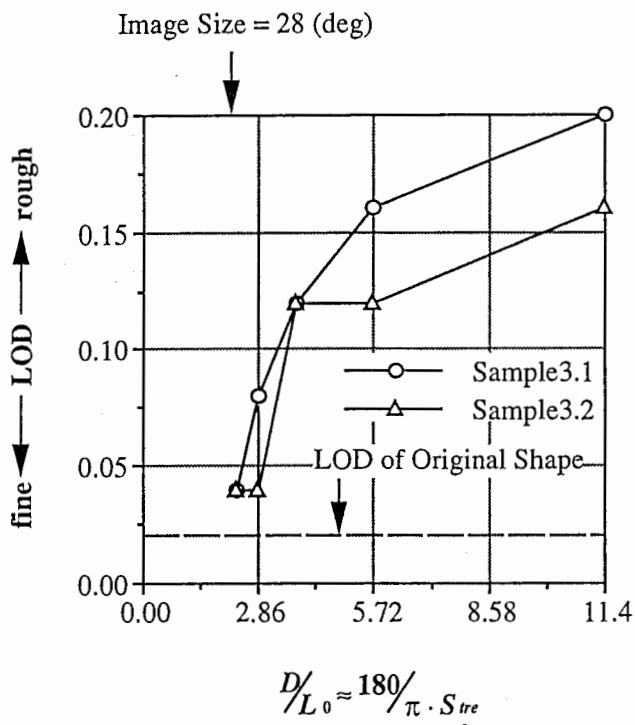


図5.10 樹木の表示位置に対する許容できる形状粗さ

そして、式(5.4)に示すように、視点からの距離に応じて形状粗さを制御する。式(5.4)は、視点から遠ざかることによる形状の簡略化の進み具合を、緩やかに抑えた制御を行う。式(5.4)で、 $a$ 、 $b$ は0より大きい適当な実数、また $Q_0$ は元の形状データの形状粗さである。

$$Q = \begin{cases} Q_0 & \left( 0 < D \leq \frac{a \cdot L_0}{b - Q_0} \right) \\ -a \cdot \frac{L_0}{D} + b & \left( \frac{a \cdot L_0}{b - Q_0} < D \right) \end{cases} \quad (5.4)$$

ここで、樹木の多数存在する仮想空間表示の際に、式(5.4)に基づいて形状データの簡略粗さの制御を行なったときの、シーン全体としてのデータ量削減の効果について考察する。簡単の為に、樹木の種類は1種類で、仮想世界に一様に密度 $\rho$ で分布しているとする。このとき、図5.11に示すように、仮想空間の小領域 $dS = D \cdot d\theta \cdot dD$ の樹木形状データ量 $dN$ は、形状粗さが $Q$ のときのデータ量を $N(Q)$ とすると、 $dN = \rho \cdot N(Q) \cdot dS = \rho \cdot N(Q) \cdot D \cdot d\theta \cdot dD$ となる。

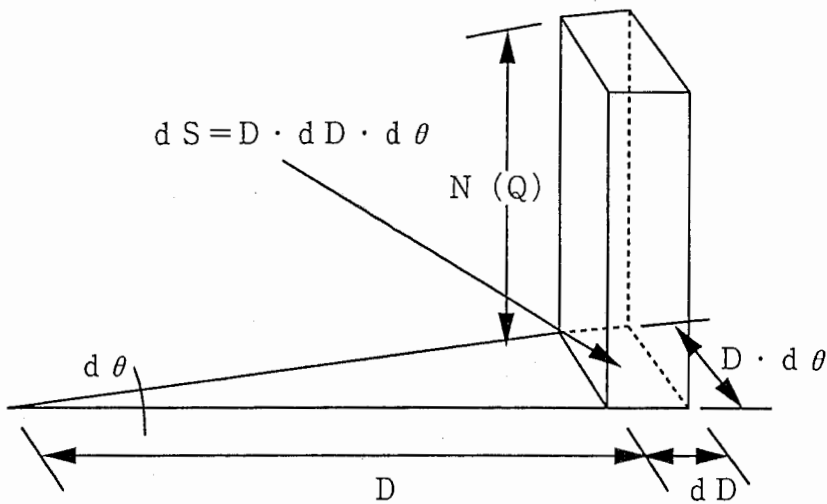


図5.11 仮想空間の小領域中の形状データ量

ここでQがDのみに依存し、 $\theta$ によらず一定であることから、視野角 $\alpha$ 中の視点からの距離xが $D \leq x < D + dD$ に存在するデータ量は、 $\alpha \cdot \rho \cdot N(Q) \cdot D \cdot dD$ と書ける。よって、図5.12に示す利用者の視野内にあたる扇形の内部の形状データ量 $N_{total}$ は、式(5.5)で計算される。式(5.5)中で、 $B = a \cdot L_0 / (b - Q_0)$ 、また、 $r$ は仮想世界の後端までの距離を表す。

$$N_{total} = \alpha \cdot \rho \cdot \left( \int_0^B N(Q_0) \cdot D \cdot dD + \int_B^r N(Q) \cdot D \cdot dD \right) \quad (5.5)$$

Qは式(5.4)に従って制御されるので、 $N(Q)$ は式(5.6)で与えられる。式(5.6)で、 $M$ はフラクタルモデルの縮小写像の個数、 $A$ はフラクタル次元を表す。

$$N(Q) = \frac{M}{M-1} \cdot Q^A = \begin{cases} \frac{M}{M-1} \cdot Q_0^A & (0 < D \leq B) \\ \frac{M}{M-1} \cdot \left( -a \cdot \frac{L_0}{D} + b \right)^A & (B < D) \end{cases} \quad (5.6)$$

式(5.4)を用いた、視点からの距離に応じた形状データの簡略化手法において、 $0 < D \leq B$ では簡略化は行なわれない。そこで、以後この領域を、利用者の近傍領域と呼ぶ。この利用者の近傍領域の表示に必要なデータ量を $N_{t1}$ とする。これは式(5.5)の右辺の第1項を計算した結果である。そして、式(5.5)の右辺の第2項を計算した結果を $N_{t2}$ とする。一方、形状簡略化を行わず表示を行なう場合に必要なデータ量は、 $N_{non} = \alpha \cdot \rho \cdot N(Q_0) \cdot r^2 / 2$ と計算される。

このとき、上記手法による形状簡略化の効果を評価するために、 $C_1(r) = N_{non} / N_{t1}$ 、 $C_2(r) = N_{total} / N_{t1} = (N_{t1} + N_{t2}) / N_{t1}$ として、3章での評価に用いた Sample3.1, Sample3.2

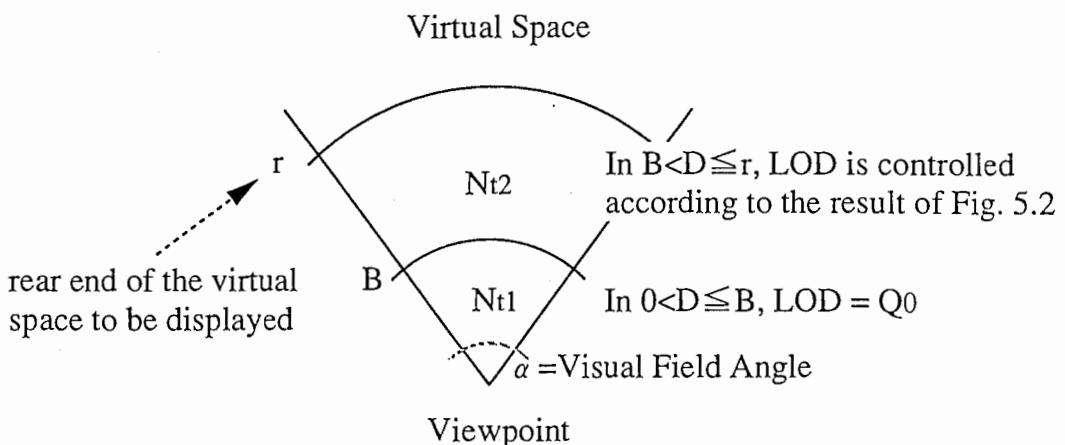


図5.12 仮想空間中の形状データの総量 ( $N_{t1} + N_{t2}$ )

についてプロットした結果を，図 5. 13 に示す．図 5. 13 の縦軸は対数表示である．また，計算に使用したパラメタを表 5. 2 に示す．

表示対象となる仮想空間の後端  $r$  が大きくなるに従い， $C1(r)$  は  $r$  の 2 乗に比例して増加する一方で， $C2(r)$  は  $r$  に対しほとんど増加しない．これは，仮想空間中に樹木が一様に分布す

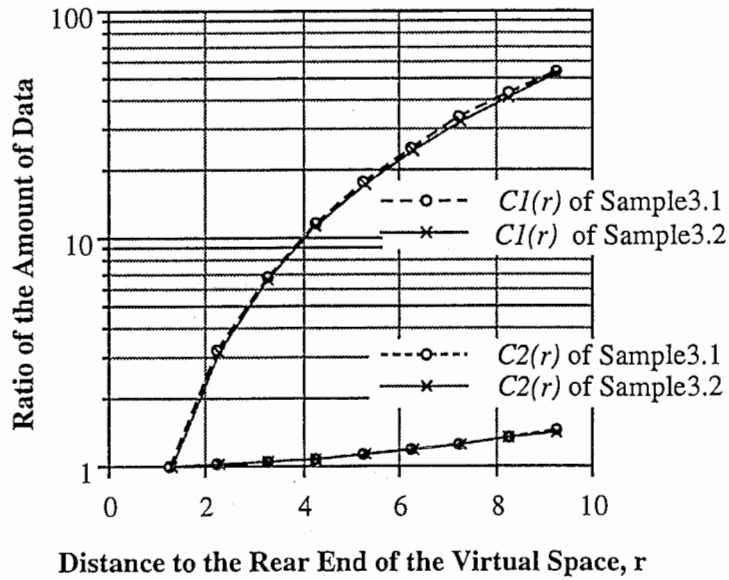


図 5. 13 フラクタルを用いた樹木形状簡略化による形状データの減少率

表 5. 2 形状データ量の計算に用いたパラメタ

	Sample3.1	Sample3.2
M	5	5
$\lambda$	0.46	0.45
A	-2.18	-2.39
$\alpha$	80(deg)	80(deg)
Q0	0.02	0.02
a	0.008	0.24
b	0.0064	0.192
L0	2000(mm)	2000(mm)

るシーンの描画に対する、フラクタルを用いた形状簡略化の効果を示している。

ここで、樹木形状の簡略化の行われぬ、利用者の近傍領域の表示に必要なデータ量について考察する。利用者の近傍領域の境界Bは、Sample3.1, Sample3.2とも、ほぼ樹木サイズL0の約2倍となる。このとき表示面には、サイズ約 $28^\circ$ の樹木画像が表示されている。臨場感のある映像を呈示するには、視野角約 $80^\circ$ が必要と言われているので[119]、利用者の近傍領域に簡略化の行われぬ樹木形状が幾つか存在することになる。

例えばSample3.2が、利用者の近傍領域に3本程度( $\sim 80\text{deg}/28\text{deg}$ )存在するとき、簡略化されていない形状データ量は約14,000ポリゴンであるので、利用者の近傍領域の表示には4万ポリゴン以上のデータ量が必要となる。臨場感通信のように、遠隔地の利用者の人物像、その他の背景物などのCG表示を行なうためには、さらに2~3万ポリゴン程度の形状データが必要があり、1シーンで6~7万ポリゴン程度のデータ量となる。これをハイエンドの3次元CG装置(約100万ポリゴン/sec)で処理しても60~70msec/フレーム程度の画像更新レートとなる。利用者に対し表示遅れを感じさせずに画像呈示するためには、1シーンを100msecで表示することが必要と報告されている[35]。このため両眼視差画像を呈示するには、50msec/フレーム[35]で表示画像を生成しなければならない。しかも実際には、表示画像更新以外にも、通信などの処理が必要であることを考慮すると、1シーンを実時間で表示することは不可能といえる。よって、表示面上の画質を均一にするように形状粗さを制御するだけでは、仮想空間表示に十分な画像更新レートが実現できる程度には、樹木形状データ量が削減できないといえる。

### 5. 3. 2 視覚特性を用いた形状粗さの制御

リアルタイムCGの分野で、(1)中心視、周辺視、(2)動態視力(=サッカード抑制)、そして(3)融像領域を考慮して、視力の低下に応じて、比較的単純な物体の形状を簡略化する3次元CG表示する手法が提案されている。そこでフラクタルを用いた樹木形状の簡略化にこれら視覚特性を利用した場合の、形状データ量の削減効果について検討した結果を示す。

#### (1) 中心視、周辺視

人間の視覚には、網膜上の視細胞の分布状態により、注視している方向付近は詳細に識別できるが、視野の周辺部分では分解能が指数関数的に低下する。前述の広視野、高精細表示装置の検討においては、注視点を中心に半径 $6^\circ$ の円内の表示は最高空間周波数を14cpd、半径 $6^\circ$ から $8^\circ$ までの領域を7cpd、半径 $8^\circ$ から $10^\circ$ までの領域を5cpd、その外側の周辺視の領域を3cpdで表

示することで、各領域の表示解像度の差を知覚しないことを主観評価により確認している。そして、これら空間周波数が、中心視における中心窩からの偏心度に対する弁別可能空間周波数測定結果と、傾向が一致していることが示されている。

そこで、フラクタルを用いた樹木形状の簡略化における、形状データの簡略化レベルの制御についても同様に、利用者の近傍領域である  $0 < D \leq B$  の範囲においては、注視点を中心に半径  $6^\circ$  以内に表示される樹木形状に対して、元形状の形状粗さである  $Q_0$  を、半径  $6^\circ$  からは、分解能  $R(\theta)$  の低下に応じた形状粗さ  $Q_0 / R(\theta)$  を与える。ただし、 $\theta$  は注視点からのずれの角度、 $0.0 < R(\theta) \leq 1.0$  である。

つぎに、上記のように、中心視、周辺視の視覚特性に基く形状粗さの制御の効果について示す。まず、周辺視における視力低下について、分解能が指数関数的に低下するという報告に基き、式(5.7)のようにモデル化されたものを使用する。式(5.7)では、 $m$  は視力低下の生じない中心視に対応する範囲であり、広視野、高精細表示で用いられたのと同じ  $6^\circ$  を使用する。また  $n$  は減衰を調節する定数であり、 $\theta = m + n$  のときに分解能  $R(\theta) = 1/e$  となる。ここで  $e$  は自然対数である。

$$R(\theta) = \begin{cases} 1 & (0 \leq \theta \leq m) \\ \exp\left(-\frac{\theta - m}{n}\right) & (\theta > m) \end{cases} \quad (5.7)$$

このとき、表示の際の視野角を  $\alpha$  とし、また仮想空間中に一様に樹木形状が分布しており、その分布の密度を  $\rho$  とすると、利用者の近傍領域である  $0 < D \leq B$  の樹木形状データ量  $N_1$  は、式(5.8)のように計算できる。

$$N_1 = \frac{\pi \cdot B^2}{180} \cdot \rho \cdot \left\{ 6 \cdot N(Q_0) + \int_6^{\alpha/2} N\left(\frac{Q_0}{R(\theta)}\right) d\theta \right\} \quad (5.8)$$

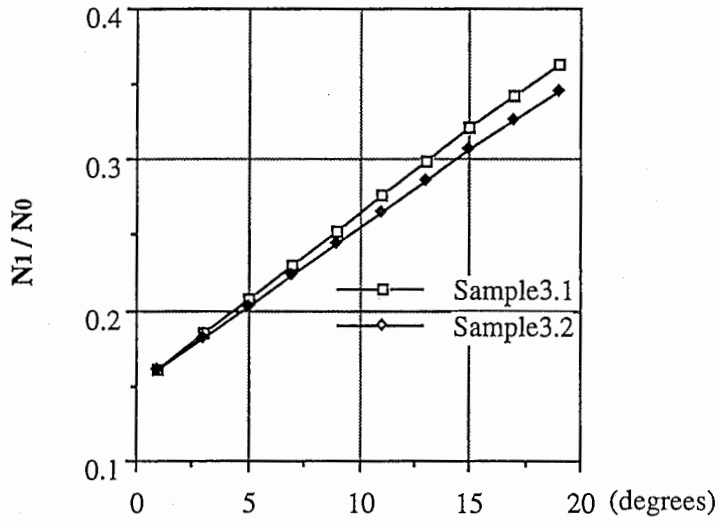
一方、中心視、周辺視を利用しないで利用者の近傍領域を表示した場合の、樹木形状データ量を  $N_0$  とすると、 $N_0$  は式(5.9)で与えられる。

$$N_0 = \frac{\pi \cdot \alpha \cdot B^2}{180 \cdot 2} \cdot \rho \cdot N(Q_0) \quad (5.9)$$

図5.14に、仮想空間の表示画角  $\alpha$  を  $80^\circ$  としたとき、Sample3.1, Sample3.2それぞれについて、 $N_1/N_0$  を  $n$  に対してプロットした結果を示す。 $n$  が大きくなるにつれて、 $N_1/N_0$  は大きくなる、すなわち形状データ量はあまり減少しない様子が分かる。また、図5.15は、 $n = 5$

°, 10°, 15° の場合の、分解能  $R(\theta)$  をプロットした結果である。

例えば、 $n = 5^\circ$  の場合では、 $R(8^\circ) = 67\%$ 、 $R(10^\circ) = 45\%$  である。これは、広視野、高精細表示において、中心視（半径  $6^\circ$  の円内）での表示の最高空間周波数 14cpd に対して、半径



1/e Point of Visual Acuity from Central Vision,  $n$

図 5. 1 4 中心視、周辺視の減衰係数に対する形状データ量の減少率

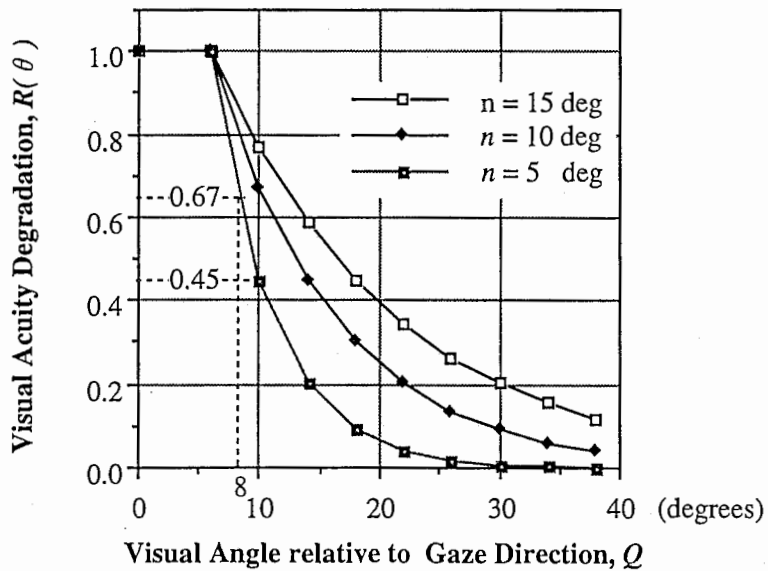


図 5. 1 5 中心視、周辺視による視力低下のモデル

6° から 8° までは 7 cpd (50%), 半径 8° から 10° までは 5 cpd (28%) で表示した, 表示画像の精細度の制御に比べて, かなり緩やかに分解能を低下である. 一方, 図 5. 14 からは,  $n = 5^\circ$  の場合に, 利用者の近傍領域のデータ量を約 20% にまで削減することができる事が分かる.

## (2) サッケード抑制

回転したり視野を素早く横切るなど, 視点に対し相対的に運動する物体については, その詳細な部分を識別する能力が低下することが知られている. これは, 眼球運動が対象の移動にスムーズに追従できなくなり, 眼球の跳躍運動, すなわちサッケードが発生することによる. 眼球が対象をスムーズに追従できる限界速度は, 20~30° /秒で, サッケード抑制による視力低下は, 約  $10^{-0.5}$  であると報告されている [88]. そして, この性質を利用したリアルタイム CG 表示についての報告がなされており [36], ドライブシミュレータなどのように, 利用者の視点が比較的高速に移動する場合には有効であると考えられている. そこで, サッケード抑制による視力低下を式 (5. 10) のようにモデル化し, データ量がどの程度削減されるかについて, 以下に示す. 式 (5. 10) では,  $\omega$  は視点に対する対象の移動速度 (角速度) を示す. また, サッケード抑制による視力低下は,  $\omega$  が 30° /秒を越えたときに発生し,  $\omega = 40^\circ$  /秒のときに視力は  $1/e$  に低下する. 各形状に与える形状粗さは, (1) 中心視, 周辺視の場合と同様に,  $Q_0/R(\omega)$  とする.

$$R(\omega) = \begin{cases} 1 & (0 \leq \omega \leq 30) \\ \exp\left(-\frac{\theta - 30}{10}\right) & (\theta > 30) \end{cases} \quad (5. 10)$$

図 5. 16 は, 図 5. 17 (a) のように視点が視線方向に移動する場合と, 図 5. 17 (b) のように視点が視線方向に垂直に移動する場合に, Sample3.1, Sample3.2 について, サッケード抑制による視力低下を考慮した形状粗さの制御を行った結果を示す. 図 5. 16 では,  $N_2$  を視力低下を考慮した場合のデータ量,  $N_0$  は視力低下を考慮しない場合のデータ量とし, その比  $N_2/N_0$  を示した. また, Sample3.1, Sample3.2 の表示サイズは 2 m とした.

図 5. 16 から, 視点が視線方向に対して垂直に移動する場合には, 例えば移動速度が 4 m/秒 (14.4 km/h) のときに, データ量を 10% 以下にすることができる. しかし, 臨場感通信のような, 他地点の利用者とのコミュニケーションを目的とする応用では, 利用者が仮想空間を駆け抜けるような状況はあまり考えられない. 成人がゆっくりと歩行する場合の速度は 1 m/秒 (3.6 km/h) 程度であり, このとき視点が視線方向に移動する場合ではデータ量は約 90% に, 視点が視線方向に垂直に移動する場合でも約 70% にしかデータ量を削減できない. 散策などの状況では, 一



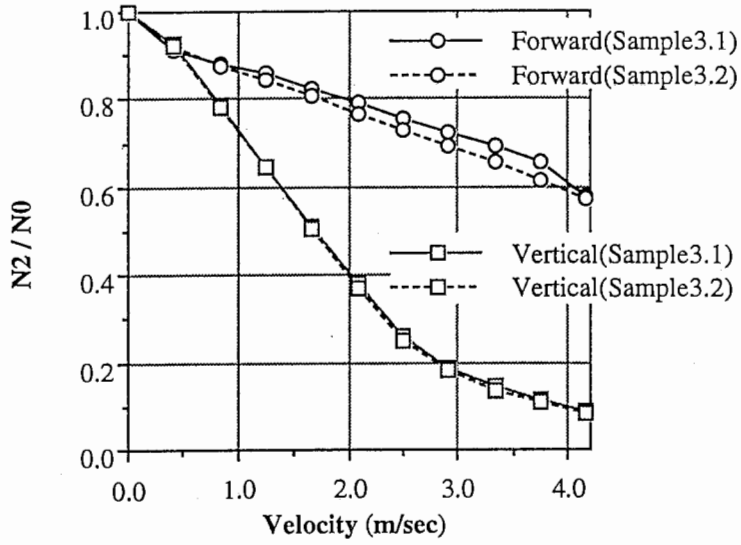


図 5. 16 サッケード抑制を用いた場合の視点移動速度に対する形状データの減少率

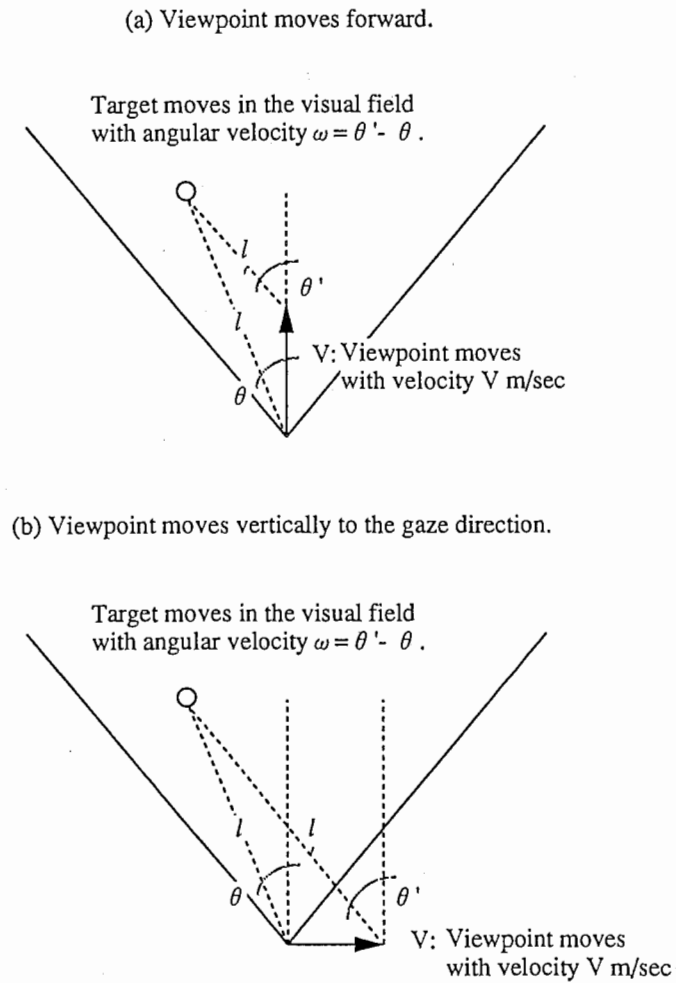


図 5. 17 視点移動に伴う仮想空間中の樹木と視点の相対速度

般に、視線方向と移動方向が全く別方向ということは少ないと思われ、このため、サッケード抑制を用いた形状簡略化の効果は、視点が視線方向に移動する場合の約90%に近いと考えられる。

### (3) 融像領域

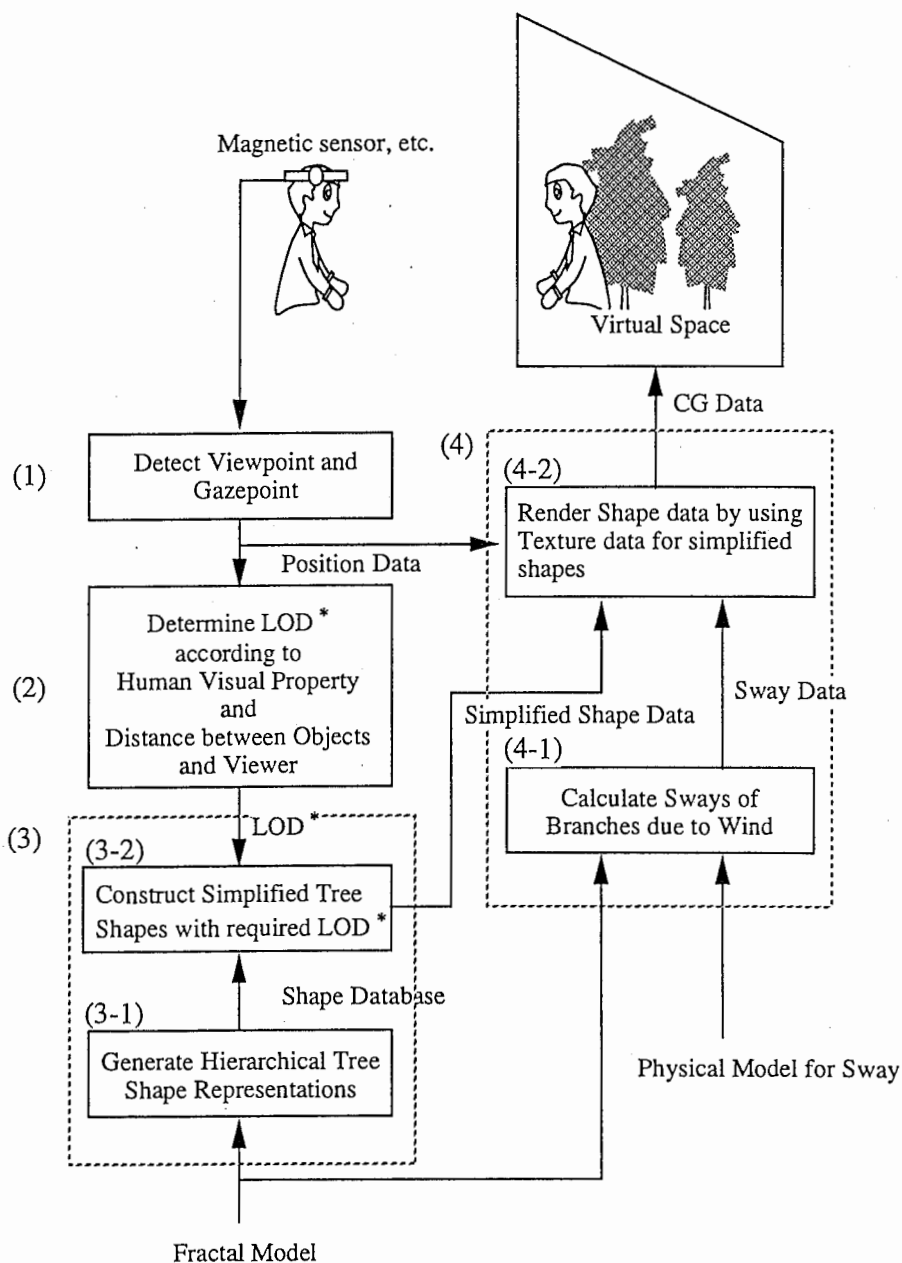
両眼立体視を行なう場合には、注視している物体の右眼像と左眼像の対応点が一致するように眼球が運動し(輻輳運動)、注視する物体が単一の立体像として知覚される(融像)。このとき、融合領域内部に存在する他の物体も融像する一方、それ以外の領域にある物体は二重像として知覚される[66][85][88]。この性質を利用して、融像していない物体は重要度が低いとし、精細度の低い形状データを用いて3次元CG表示する方法が提案されている[36]。

しかし、二重像として知覚される場合に失われる情報は両眼視差情報であり、正確な奥行き情報が失われただけで、観察される画像の画質が低下している、すなわち視力低下が生じているとは考えにくい。一方、眼球の輻輳運動は焦点調節機能とは密接に関係しており、輻輳が正しく合っていないため融像できない場合には、焦点が合わないことによる視力低下が生じる可能性はある。しかし一般に、我々は日常生活において二重像を意識することはなく、視力感度の高い中心視領域では、通常、輻輳も焦点調節も正しく行われていると考えられる。よって、融像領域による視力低下については考慮しない。

以上のことから、本研究では、中心視、周辺視の視覚特性のみを考慮して、フラクタルを用いた形状簡略化に与える形状粗さを、中心視から緩やかに分解能を低下させるよう制御する。これにより、利用者の近傍に存在する樹木形状データ量を、20%程度に削減できると予測される。その結果、利用者の近傍の樹木形状データ量の約3万ポリゴンを、6千ポリゴン程度に削減し、遠隔地の利用者の人物像、その他の背景物などの形状データ、2万ポリゴンと合わせても、仮想空間表示に必要な画像更新レート(50msec/フレーム)が達成できるといえる。

## 5. 4 樹木の多数存在する仮想空間表示への適用

本節では、(1)フラクタルを用いた樹木形状の簡略化手法、(2)テクスチャマップを用いた3次元樹木画像の生成方法、そして、(3)中心視、周辺視の性質を利用した形状粗さの制御方法を組み合わせた仮想空間表示評価システム(図5.18)の概要について述べる。



LOD\*: Level of Details

図5.18 樹木の多数存在する仮想空間表示評価システムの概要

## 5. 4. 1 樹木の多数存在する仮想空間表示評価システムの概要

図5.18に示した仮想空間表示評価システムの各サブシステムについて、その概要を説明する。

### (1) 視点位置、および視線の検出

臨場感通信においては、利用者の視点位置を磁気センサーを用いて実時間で検出しているが、視線位置については、画像処理により瞳孔位置を検出するなどの処理が必要となる。しかし仮想空間表示には液晶シャッター方式を使用しているため、利用者は特殊な眼鏡を着用する必要があり、現状では視線位置検出はできない。そこで視線位置は表示画面中央に固定とした。また、視点位置移動については、マウス入力により仮想空間中を移動するようにした。今後、特殊眼鏡無しでの立体表示、あるいは特殊眼鏡を掛けたままでの視線位置検出が可能になった場合には、実際に視点位置、そして視線の検出を行う。

### (2) 形状粗さの制御部

本処理では、利用者の視点位置を入力とし、式(5.4)で示した主観評価実験結果に基づくモデルを用い、視点からの距離に応じて、簡略化形状の形状粗さを定める。さらに、人間の視覚特性、すなわち網膜上の視細胞の分布状態により、注視している方向付近は詳細に識別できるが、視野の周辺部分では分解能が指数関数的に低下する性質も利用する。

### (3) フラクタルを用いた簡略化形状の生成部

本処理では、3章で述べたように、表示に前もって樹木形状のフラクタルモデルに基づき簡略化形状を生成する。そして、高速に簡略化形状を演算するため、簡略化形状の多重化を行う。また、5.2.3で述べた、簡略化形状にテクスチャマップするためのテクスチャデータを生成する(3-1)。そして、表示に際しては、観察者の視点位置、視線方向に応じて定められた形状粗さを元に、簡略化された樹木形状を生成する(3-2)。

### (4) 簡略化形状を用いた3次元樹木画像生成部

#### (4-1) 風などの外力による樹木の揺らぎの計算

本処理では、簡略化形状を用いて表示される樹木形状に対して、風などの外力による変形のデータを計算する。樹木の形状は、一般に極めて複雑であるので、樹木の枝や葉の風などの外力による変形の、正確な物理的なモデルを構築することは困難である。そこで、枝の振動モード解析に基づ

くモデル [46, 47], あるいは粒子によるシミュレーションを樹木形状に適用した例 [48] などが報告されている。

本処理では, 形状の変形計算を更に単純化するため, 図 5. 19 に示すように, 樹木の各枝がばねにより連結されているモデルを用いた。また, 各枝に働く力は風などによる外力のみとし, 枝間の力の相互作用は考慮していない。各枝は, 固有振動数  $f$ , 外力による釣り合いの位置の角度変位  $\theta_0$ , 振幅  $A$  をパラメタとして持つ。

フラクタル樹木形状は, 最初の幹に再帰的に縮小写像を適用することで得られるので, 枝の世代が進むにつれて枝の長さは短くなる。よって枝の振動数  $f$  は, 世代が進むほど高くなるよう設定する。また, 枝は通常先端に向かうほど柔らかくなる。そこで, 枝のたわみに相当する  $\theta_0$ , 揺らぎの振幅  $A$  についても, 世代が進むほど大きくなるよう設定する。

以下に, 樹木形状の揺らぎによる変形の計算方法を示す。

- (a) 時刻  $t = 0$ , 枝の世代  $g = 0$ , たわみの履歴行列  $H$  を単位行列とする。
- (b) 枝の始点  $S$  と終点  $E$  をたわみの履歴行列  $H$  により変換し,  $S' = H \times S$ ,  $E' = H \times E$  を計算する。
- (c) 風向に対する, 第  $g$  世代の枝の時刻  $t$  でのたわみ角  $\theta_g$  を,  $\theta_g = A_g \cdot \sin(2\pi \cdot f_g \cdot t) + \theta_{0g}$  にて計算する。  $f_g$ ,  $A_g$ ,  $\theta_{0g}$  は, 第  $g$  世代の各枝のパラメタである。

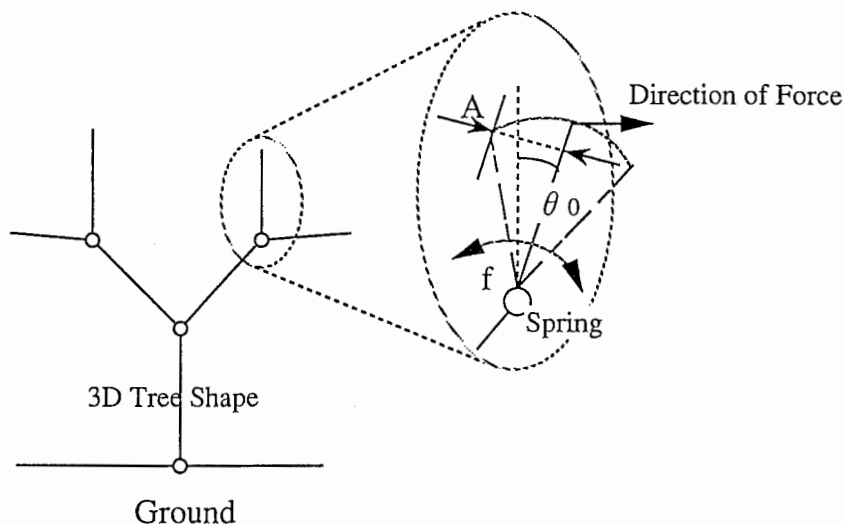


図 5. 19 風による揺らぎを計算するための単純な物理的モデル

- (d) たわみは、 $S'$  と  $E'$  を結ぶベクトルと、風向のベクトルで張られる平面上で生じるように、たわみの座標変換行列  $H_g$  を計算し、たわみの履歴行列  $H = H_g \times H$  とする。また、たわみ後の枝の終点  $H_g \times E'$  を計算する。
- (e)  $g$  を次の世代として (2) から処理を行なう。
- (f) 全ての世代について (b) から (e) の処理が終了すれば、時刻  $t = t + \Delta t$  として、(b) から処理を行なう。

#### (4-2) 簡略化形状用テクスチャデータを用いたレンダリング

本処理では、5.2.3 で述べたように、指定された形状粗さにより生成された、簡略化された樹木形状データに対して、細部の形状や色調を与えるよう、簡略化形状用テクスチャをテクスチャマップする。

### 5.4.2 評価結果

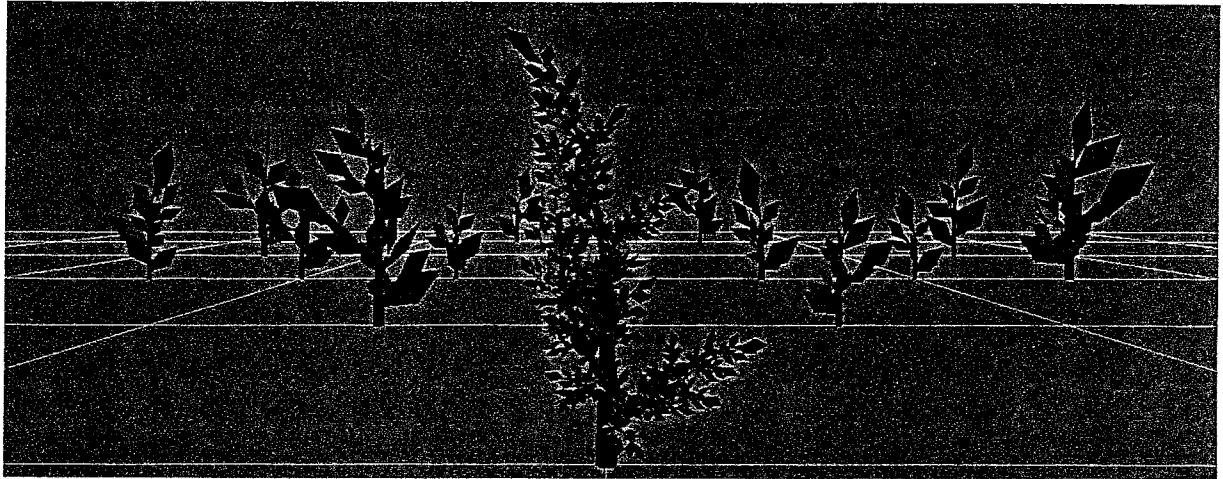
ここでは、前述の仮想空間表示評価システムの効果を確認した結果について、表示例とデータ量削減効果について示す。画像生成に使用した計算機は、Silicon Graphics 社の Onyx/4 RE2 (性能 160 万 Tmesh/秒) を用いた。ただし、この性能は面積 50 ピクセルの 3 角形メッシュを用い、テクスチャマップ有、光源計算なし、フラットシェーディングなどの条件下で測定されたものである。

図 5.20 (a) は、フラクタルを用いた高速 3 次元樹木画像生成法を用いた表示を行なった結果である。ここでは、中心視、周辺視の視覚特性は利用していない。表示には、Sample3.1 の樹木形状データを用いた。視点から遠ざかるにつれて、樹木形状を簡略化していく様子が分かる。これに簡略化形状用のテクスチャデータを用いた表示を行なうと、図 5.20 (b) のような表示結果になり、利用者に簡略化した形状を意識させない表示が可能になる。図 5.20 の表示例では、フラクタルを用いた高速 3 次元樹木画像生成法を用いずに表示した場合の全体の形状データ量、52,365 ポリゴンに対し、使用した場合の形状データ量は 4,025 ポリゴン (約 8%) まで減少した。Sample3.1 の表示に使用した簡略化形状の多重化表現のデータサイズは、3 章で示したのと同じ、約 98K byte である。また簡略化形状用のテクスチャデータサイズは約 4M byte であり、すべての樹木で共有して使用している。このときの画像生成時間は 48msec (形状演算時間 14msec, 描画時間 34msec) であった。

図 5.21 (a) は、フラクタルを用いた高速 3 次元樹木画像生成法に、中心視、周辺視の視覚特性を用いた形状粗さの制御を組み合わせ表示した結果である。本表示例では、視線は表示面の中

央に向いているとした。表示には、Sample3.1の樹木形状データを用いた。図5.20と同様、視点から遠ざかるにつれて、樹木形状を簡略化していく様子が分かる。さらに表示面の周辺、すなわち周辺視の領域で、形状が簡略化されている。図5.21(b)は、簡略化形状用のテクスチャデータを用いて表示した結果である。図5.21の表示例では、中心視、周辺視の視覚特性を用いずに表示した場合の全体の形状データ量、7,809ポリゴンに対し、使用した場合の形状データ量は4,049ポリゴン(約52%)に減少している。Sample3.1の表示に使用した簡略化形状の多重化表現のデータサイズは、3章で示したのと同じ、約98K byteである。また簡略化形状用のテクスチャデータサイズは約4M byteであり、すべての樹木で共有して使用している。このときの画像生成時間は48.6msec(形状演算時間14.3msec、描画時間34.3msec)であった。

また、図5.22に、樹木が多数存在する場合で、揺らぎによる変形を伴う情景の3次元画像を生成した例を示す。図5.22は両眼視差画像であり、交差法により立体視することができる。



(a) Before texture-mapping



(b) After texture-mapping

図 5. 2 0 フラクタルを用いた形状簡略化による表示例





(a) Before texture-mapping



(b) After texture-mapping

図 5 . 2 1 中心視，周辺視を用いた形状粗さの制御による表示例

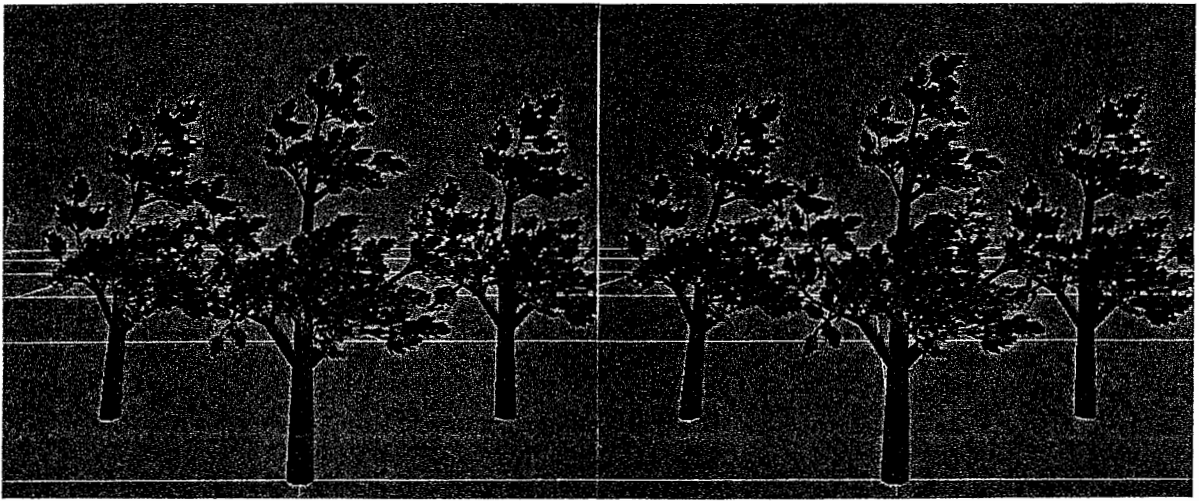


図5. 22 樹木の多数存在するシーン（風による揺らぎ）の表示例（交差法）

## 5. 5 まとめ

本章では、まず、フラクタルを用いて簡略化された樹木形状を用いて3次元樹木画像を生成する際に、簡略化形状に対し樹木画像をテクスチャマップすることで、形状を大きく簡略化しても、利用者にそれを意識させない表示が可能であることを、主観評価実験により明らかにした。そして、この考え方による、簡略化形状用のテクスチャデータを用いたレンダリング方法を提案した。

また、実時間CG表示が可能な程度にシーン全体の樹木形状データを削減するためには、シーン中の樹木それぞれに与える形状粗さを制御する方法を検討する必要がある。そこで、樹木の多数存在する仮想空間を表示する場合に、利用者の視点位置などを検出し、それをもとに中心視/周辺視といった視覚特性を利用して、表示面上で重要な部分のみを高品質に表示することで、表示に必要な形状データ量が大幅に減少することを示した。

つぎに、(1) フラクタルを用いた樹木形状の簡略化手法、(2) テクスチャマップを用いた3次元樹木画像の生成方法、そして、(3) 中心視、周辺視の性質を利用した形状粗さの制御方法を組み合わせた仮想空間表示評価システムを実装し、その効果を確認した。

以下に、それらの結果をまとめる。

### (1) テクスチャマップを用いた簡略化形状の表示方法について

樹木画像をテクスチャマップすることにより、簡略化した形状データを用いて3次元樹木画像を表示した場合にも、違和感のない表示が可能であることが分かった。そして、この考え方による、簡略化形状用のテクスチャデータを用いたレンダリングにより、樹木の揺らぐ画像を生成した結果、変形のない状態では見えなかった部分が露呈する場合でも、違和感のない表示が行なえることを示した。

### (2) 視覚特性を用いた形状粗さの制御について

シミュレーションによる評価の結果、自然の景観のように樹木の多数存在するシーンを、実時間で3DCG表示するためには、フラクタルを用いた樹木形状の簡略化と、視覚特性を用いた形状粗さの制御を組み合わせ、シーン全体の樹木形状データを削減させる必要があることが分かった。

つぎに、視覚特性を用いて簡略化形状の形状粗さを制御した場合の、形状データの削減効果について、中心視、周辺視を用いた場合、20%にまで形状データを削減することができることが分かった。一方、臨場感通信のように、遠隔地の利用者同士の面談を目的とするような場合を想定すると、動態視力を用いた場合には約90%にまでしか形状データ量を削減できないことが分かった。また、

融像領域の利用については、視力低下に直接関連するとは考えにくいと結論した。

### (3) 樹木の多数存在するシーンを表示例について

本章で示した表示例では、フラクタルを用いた高速3次元樹木画像生成法を使用した場合の形状データ量は、これを用いずに表示した場合の約8%まで減少した。また、中心視、周辺視の視覚特性を用いて形状粗さを制御した場合、形状データ量は、さらに約52%に減少した。このようにシーン全体の形状データ量を減少させることで、風により多数の樹木が揺らぐようなシーンも、実時間で3DCGにより表示可能になった。

## 6. 結論

本論文は、臨場感通信などの人工現実感の応用において、今後、複雑化、高度化していく仮想空間を3DCGを用いて表示する際に、十分な視覚的臨場感と対話性を実現する手法に関する研究をまとめたものである。

具体的には、自然の造形物として特に樹木を対象とし、これを多く含む仮想空間を3DCGを用いて表示する場合に、十分な視覚的臨場感と対話性を実現するため、利用者に画質の劣化を感じさせることなく、シーンの描画に必要なデータ量を、実時間表示可能な程度に削減可能にする手段を提供する。

以下に研究成果を要約し、さらに今後の課題・展望について述べる。

まず、樹木を対象を多数含む仮想空間を表示する場合について、仮想空間の視覚的臨場感、対話性を実現するための重要な技術である3次元画像生成技術について、高速画像生成、そして形状モデリングという観点から、従来の研究を分類し説明した。高速画像生成の観点からは、樹木形状のような複雑な形状を効果的に簡略化する手法が必要であること、そして形状モデリングの観点からは、実際の樹木に似せたリアリティの高い形状を容易に生成する手法が必要であることを述べた。従来の研究では、画像生成過程、モデリング過程それぞれにおいてこれら問題が個別に検討されてきたが、本研究の対象とする樹木のような複雑形状に対しては、モデリング過程では実時間性を考慮した複雑形状モデルを生成し、高速画像生成過程ではその複雑形状モデルの特徴を利用した形状簡略化を行うというような、モデリング過程から高速画像生成過程までの全体の枠組みの中で問題解決方法を検討することが重要であることを述べた。

そして、本研究では、3章において、高速画像生成を可能にする以下の方法について述べた。

- (1) 樹木のような複雑な形状を効果的に簡略化するフラクタルを用いた樹木形状簡略化、及び表示の際の簡略化形状の演算時間を短縮する簡略化形状の多重化を提案し、その有効性を確認した。

さらに、5章において、簡略化形状を用いても、観察者にとって違和感のない表示を可能にする

以下の方法について述べた。

- (2) 簡略化形状を観察者に意識させることのない表示方法として、簡略化形状用のテクスチャを用いた表示を提案した。
- (3) 形状簡略化のレベルを中心視、周辺視といった視覚特性を利用して制御することで、自然景観のような樹木が多数存在するシーンに含まれる樹木形状データ量を、実時間で3DCG表示可能な程度にまで削減できることを示した。

また、4章において、実際の樹木に似せたりアリティの高い形状を容易に生成する以下の手法について述べた。

- (4) 上記の形状簡略化に用いるフラクタルモデルを、実写樹木のシルエット画像から自動的に推定する方法を提案し、実際の景観画像中の樹木に似せた形状を、仮想空間中に表示できることを示した。

つぎに、残された課題について述べる。

#### (1) 複雑形状のモデルとしてフラクタルを用いる限界

樹木形状のような自然の複雑形状は、統計的にフラクタル性を有しているとされているが、その形状は、本論文で用いた、IFSのみによるフラクタルモデルに合致しているとは限らない。それは、樹木の遺伝子にプログラムされた成長のルールは規則的であっても、実際の樹木形状は、重力や日当たりなどの成育環境の影響で必ずしも自己相似の性質を有さないからである。樹木形状を忠実に再現しようとするれば、やはり成長モデルに様々な要因を付与してモデルを精密化する必要になるが、これは、本論文の2章「仮想空間表示における課題と本研究での取り組み」で述べたように、実時間性、あるいは形状の再構成の容易さとの二律背反となる。

そこで今後、フラクタルの複雑形状を有効に簡略化する性質を利用しつつ、かつ、複雑形状に対するモデリング能力を強化するため、IFSのみに基づくモデルにランダム性を付与した場合の形状簡略化、あるいは実写樹木画像からのモデルの再構成について検討を行なう。

## (2) 樹木形状以外の複雑形状物体への適用

本論文で得られた、フラクタルを用いた形状簡略化、そしてダイナミックな動きの存在するシーンの実時間画像生成に関する知見は、フラクタルで近似ができる形状一般について、例えば雲、煙、炎などといったものにも適用できると考えられる。

本論文では、3章で述べたように、形状簡略化、あるいは形状再構成に使用するフラクタルモデルを、樹木形状を生成するのに適するようにパラメタに制約を課した。また、フラクタルモデルから形状データを生成する際にも、樹木の成長を模した、親の枝から子の枝が派生する手法を用いた。そこで、雲、煙、炎などに対しても、これらを記述するのに適当なフラクタルモデル、そして形状生成法について検討し、本論文で提案した手法を適用し、評価を行なう。

## (3) 実写画像中からのフラクタル形状の自動的なセグメンテーション

高度化、複雑化する仮想空間の設計の労力を軽減するためには、実写の景観画像から、仮想空間表示に用いる景観のモデルを自動的に生成できることが理想である。このためには、実写景観画像中の物体を自動的に分離する技術が必要である。

実写画像から樹木領域を切り出すことは、フラクタル次元を用いた画像解析 [120] [121]、あるいは色相、テクスチャなどの複数の特徴を用いた領域分割法 [122] によりある程度は可能であることから、将来的には、自然景観画像から、樹木などのモデリングを自動的に行なうことが可能になると期待できる。

## (4) リアリティの高い揺らぎの生成

リアリティの高い揺らぎの画像を生成するためには、例えば、時事刻々変化する風の動きに対応して、樹木の枝のなびく方向が変化するといったことが必要と考えられる。このためには、仮想空間全体に渡っての風向、風力の制御と、それらに応じた実時間での揺らぎデータの生成を行なわなければならない。

本論文で述べた、単純な物理モデルに基づく揺らぎデータの生成は、現状では実時間での揺らぎの計算は実現できていない。しかし、枝の振動モード解析に基づくモデル [46, 47]、あるいは粒子によるシミュレーションを樹木形状に適用した例 [48] などと異なり、枝間の力の相互作用を考慮していないため、処理の並列化は非常に容易である。従って、今後、並列処理を利用した実時間での揺らぎデータの生成について検討する。

本論文の序論で述べたように、近い将来、B-ISDNの普及に伴い、ネットワークリアリティと呼ばれる、高速ネットワーク上に構築された仮想的なコミュニティ空間の生成に、人工現実感の技術が適用されていくであろう。また、建築シミュレーション、医療などの分野でも、これまで以上にリアリティの高い仮想空間を生成することが必要になってくるであろう。本論文で提案した、自然景観を対象とするような、高度化、複雑化した仮想空間を表示する技術が、様々な分野での人工現実感の応用、発展の一助になれば幸いである。



## 参考文献

- [1] 舘暲：“テレイグジスタンス”，日本ロボット誌，4，3，pp.295-300（1986）。
- [2] 服部桂：“人工現実感の世界”，（株）工業調査会，（1991）。
- [3] M.W.Krueger 著／下野隆生訳：“人工現実”，（株）トッパン，（1991）。
- [4] 舘暲・廣瀬通孝監修／著：“バーチャル・テック・ラボ”，（株）工業調査会，（1992）。
- [5] 廣瀬通孝，中垣好之：“バーチャルリアリティ技術と仮想世界”，光技術コンタクト，31，11，pp.637-642（1993）。
- [6] 伴野明，岸野文郎：“臨場感通信会議におけるヒューマンインターフェイス技術”，人工知能学会誌，6，3，pp.358-369（1991）。
- [7] 寺島信義：“臨場感通信会議システムの構築に向けて”，画像ラボ，1，pp.46-48（1993）。
- [8] F.Kishino, J.Oya, H.Takemura, N.Terashima：“Virtual Space Teleconferencing System - Real Time Detection and Reproduction of 3D Human Images”，HCI'93 Proceedings，（1993）。
- [9] H.Noma, Y.Kitamura, T.Miyasato, F.Kishino：“Multi-Point Virtual Space Teleconferencing System”，IEICE Trans. on Communications, E78-B, 7, pp.970-979（1995）。
- [10] D.Zeltzer：“Autonomy, Interaction, and Presence”，Presence：Teleoperators and Virtual Environment to appear，
- [11] 斎田真也：“眼球運動の計測”，製科研報告，99，pp.20-26（1984）。
- [12] L.E.Sutherland：“A Head Mounted Three Dimensional Display”，Fall Joint Computer Conf.，pp.757-764（1968）。
- [13] S.S.Fisher, M.McGreevy, J.Humphries, W.Robinette：“Virtual Environment Display System”，Proceedings of 1986 ACM Workshop on Interactive Graphics，pp.77-87（1986）。
- [14] 廣瀬通孝，佐藤慎一：“人工現実感技術を用いた臨場感の合成とその伝達”，Proceedings of 9th Symposium on Human Interface，pp.111-116（1993）。
- [15] 廣瀬通孝，中垣好之，谷口雅昭，小森信也，遠藤隆明：“はいれるテレビの開発”，テレビ学技報，18，41，pp.1-8（1994）。
- [16] 竹村治雄，伴野明：“時分割両眼立体視を用いた3次元指示入力操作の評価”，信学論（D-I），J74-D-I，1，pp.33-38（1991）。
- [17] 高橋友一，岸野文郎：“手振り認識方法とその応用”，信学論（D-II），J73-D-II，12，pp.1985-1992（1990）。

- [18] 竹村治雄, 岸野文郎 : “人工現実感によるヒューマンインターフェース”, テレビ誌, 44, 8, pp.981-985 (1990) .
- [19] 佐藤誠, 平田幸広, 河原田弘 : “空間インタフェース装置 SPIDAR の提案”, 信学論 (D-II), J74-D-II, 7, pp.887-894 (1991) .
- [20] 石井郁夫, 柄沢達明, 牧野秀夫 : “触覚機能を持つ仮想物体操作システム”, 信学論 (D-II), J76-D-II, 9, pp.1860-1867 (1993) .
- [21] 秋山宣英, 林正樹, 山内結子, 小林睦朗, 佐藤正明 : “NHK スペシャル「ナノスペース」におけるリアルタイムCGと実写映像の合成技術”, テレビ学技報, 17, 23, pp.13-18 (1992) .
- [22] 林正樹, 山内結子, 福井一夫, 榎並和雅 : ““仮想スタジオ””, テレビ学技報, 16, 10, pp.13-18 (1992) .
- [23] 山内結子, 林正樹, 福井一夫, 榎並和雅, 坂井滋和 : “電子スタジオシステム～番組「ナノスペース」への適用”, テレビ学技報, 17, 23, pp.1-6 (1992) .
- [24] 井上誠喜, 柴田正啓 : “シーン合成用画像データベース”, 画像電子学会研究会予稿, 89-02-06, pp.32-37 (1989) .
- [25] 井上誠喜, 柴田正啓, 中須英輔 : “画像合成のための部品画像ファイルシステム”, 信学論 (D-II), J72-D-II, 11, pp.1824-1832 (1989) .
- [26] 田中弘美, 岸野文郎 : “曲面再構成のための適応格子生成法 - 階層的三角パッチ表現並列生成アルゴリズム -”, 信学論 (D-II), J76-D-II, 8, pp.1691-1703 (1993) .
- [27] N.M.Thalmann, D.Thalmann : “The Artificial Life of Synthetic Actors”, 信学論 (D-II), J76-D-II, 8, pp.1506-1514 (1993) .
- [28] 鶴沼宗利, 武内良三 : “コンピュータアニメーションにおける感情を伴った人間の歩行動作の生成方法”, 信学論 (D-II), J76-D-II, 8, pp.1822-1831 (1993) .
- [29] 梶原創一 : “物理的定義に基づいた時変ホモトピーを用いた3次元顔画像の表情と動作の生成”, テレビ誌, 49, 3, pp.344-356 (1995) .
- [30] 中津良平 : “知能映像通信研究所の研究内容の紹介”, ATR Journal, 19, pp.7-10 (1995) .
- [31] 舘暉 : “ここまできたバーチャルリアリティ”, 信学誌, 77, 988-991, pp.6-9 (1994) .
- [32] 二瓶健次, 白川公子, 吉川泰弘, 唐沢豊, 久保田靖夫 : “DVIシステムを用いた仮想現実体験ビデオ、“動物園へ行こう”の長期入院する小児難病患者への応用”, 3D映像, (1993) .
- [33] 木島竜吾, 廣瀬通孝 : “仮想箱庭の開発”, Human Interface N&R, 8, pp.145-150 (1993) .

- [34] 吉田展康, 他 13 名 : “電子情報ネットワーク上に展開され、仮想現実をインターフェイスとする新しい医療空間 (Hyper Hospital) のシステム設計と試作”, *Human Interface N&R*, 9, pp.313-318 (1994) .
- [35] 佐藤隆宣, 伴野明, 岸野文郎 : “運動視における表示遅れの影響と高速画像生成表示方法”, *テレビ誌*, 46, 9, pp.1179-1188 (1992) .
- [36] 大島登志一, 内山晋二, 山本裕之, 田村秀行 : “視覚特性を利用した 3 次元グラフィックオブジェクトの適応的表示法”, 3 次元画像コンファレンス '94 講演論文集, pp.201-206 (1994) .
- [37] 田村秀行, 北村素子 : “3 次元仮想空間生成・表示へのモデル系 & 非モデル系アプローチ”, *Human Interface*, 9, pp.279-286 (1994) .
- [38] 加藤伸子, 岡崎彰夫 : “形状簡略化に基づく 3 次元オブジェクト空間の最適高速表示”, *信学論 (D-II)*, J76-DII, 8, pp.1712-1721 (1993) .
- [39] 北嶋克寛, 遊佐洋子 : “リアルタイム景観シミュレータのための形状グルーピングと多重表現に基づく描画時間の短縮”, *信学論 (D-II)*, J77-D-II, 2, pp.311-320 (1994) .
- [40] 秋葉澄伸, 山本強, 青木由直 : “都市景観デザインへのテクスチャマッピング技法の応用”, *NICOGRAPH 論文コンテスト*, pp.245-252 (1987) .
- [41] 田北晋, 西田友是, 中前栄八郎 : “リフレクションマッピングを考慮した 3 次元画像の構成法”, 第 35 回情処全大, pp.2303-2304 (1987) .
- [42] 金田和文, 秋信俊男, 中前栄八郎 : “写真画像を用いた樹木の表示法”, 第 39 回情処学全大, 39, 2, pp.907-908 (1989) .
- [43] S.Takita, K.Kaneda, T.Akinobu, H.Iriyama, E.Nakamae, T.Nishida : “A Simple Rendering for Renumbra Caused by Sunlight”, *CGI'90 Proceedings*, pp.187-201 (1990) .
- [44] K.Tadamura, K.Kaneda, E.Nakamae, F.Katoh, T.Noguchi : “A Display Method of Trees by Using Photo Images”, *Journal of Information Processing*, 15, 4, pp.526-534 (1992) .
- [45] 中前栄八郎, 多田村克己, 金田和文 : “特集 CG による自然物と自然現象の表現 テクスチャマッピングを利用した簡易樹木表示法”, *PIXEL*, 125, pp.33-38 (1993) .
- [46] 新谷幹夫 : “風による草木の運動モデル”, 第 43 回情処全大, pp.3U-8 (1991) .
- [47] M.Shinya, A.Fournier : “Stochastic Motion - Motion Under the Influence of Wind”, *EUROGRAPHICS'92*, 11, 3, pp.C119-C128 (1992) .

- [48] 千葉則茂, 河野充, 佐藤義人, 村岡一信, 斎藤伸自 : “風による樹木の揺らぎ画像生成法の検討”, 画電学誌, 22, 5, pp.475-483 (1993) .
- [49] H. Peitgen, D. Saupe : “The Science of FRACTAL IMAGES” , Springer-Verlag, (1988) .
- [50] B. B. Mandelbrot : “THE FRACTAL GEOMETRY OF NATURE” , Freeman, (1977) .
- [51] M. F. Barnsley : “Fractals Everywhere” , Academic Press, (1988) .
- [52] H. Peitgen, H. Jurgens, D. Saupe : “Chaos and Fractals, New Frontiers of Science” , Springer-Verlag, (1993) .
- [53] 桑原教彰, 志和新一, 新井民夫 : “樹木画像を入力とする3次元樹木形状のフラクタルモデルの自動推定方法” , 画電学誌, 24, 5, pp.541-549 (1995) .
- [54] 桑原教彰, 志和新一, 岸野文郎, 新井民夫 : “樹木画像を入力とする3次元樹木形状のフラクタルモデルの自動推定方法の評価” , 画電学誌, (採録) .
- [55] 杉江日出澄, 岡崎明彦, 足達義則, 尾崎正弘 (共著) : “FORTRAN77による数値計算法” , 培風館, (1986) .
- [56] 桑原教彰, 鉄谷信二, 志和新一, 岸野文郎 : “フラクタルを用いた階層的な樹木形状表現による3次元樹木画像の高速生成方法” , 信学論 (D-II), J78-D-II, 7, pp.1091-1104 (1995) .
- [57] A. R. Smith : “Plants, Fractals, and Formal Languages” , Computer Graphics, 18, 3, pp.1-10 (1984) .
- [58] E. Oppenheimer : “Real Time Design and Animation of Fractal Plants and Trees” , Computer Graphics, 20, 4, pp.55-64 (1986) .
- [59] 本多久夫 : “樹木の幾何学的模型” , 数理科学, 184, pp.18-25 (1978) .
- [60] 本多久夫 : “樹木の分岐” , 数理科学, 221, pp.51-57 (1981) .
- [61] 本多久夫 : “樹木の三次元モデル” , Science on Form, 1, 1, pp.85-94 (1985) .
- [62] Y. Kawaguchi : “A Morphological Study of the Form of Nature” , SIGGRAPH '82, pp.223-232 (1982) .
- [63] P. Prusinkiewics, A. Linder Mayer, J. Hanan : “Development Models of Herbaceous Plants for Computer Imagery Purpose” , Computer Graphics, 22, 4, pp.141-150 (1988) .
- [64] クリフォード A. ピックオーバ : “植物の形状と成長をシミュレーション” , 日経CG, 10月号, pp.107-110 (1987) .

- [65] N.Kuwahara, S.Shiwa, F.Kishino : “A Method for Displaying Virtual Spaces of Natural Scenes Employing Fractal-based Shape Data Simplification and Visual Properties”, IEICE Trans. on Information and Systems, (採録) .
- [66] 畑田豊彦 : “立体映像の視覚心理”, 映像情報, 20, 1, pp.27-36 (1988) .
- [67] 畑田豊彦, 斎田真也 : “奥行き知覚の要因とメカニズム”, テレビ誌, 43, 8, pp.755-762 (1989) .
- [68] 長田昌次郎 : “両眼立体画像観視における両眼融合限界の画角および視距離依存特性”, テレビ誌, 43, 3, pp.276-281 (1989) .
- [69] 鉄谷信二 : “3次元表示技術:眼鏡なし立体表示”, VISION, 5, pp.33-40 (1993) .
- [70] 平田渥美, 鈴木清明 : “3次元ディスプレイ”, テレビ誌, 41, pp.610-618 (1987) .
- [71] 濱崎襄二 : “レンチキュラ画像”, 画電学誌, 10, pp.415-428 (1981) .
- [72] 鈴木清明 : “立体ビデオディスクについて”, テレビ学技報, VR87-2, pp.7-12 (1987) .
- [73] 永嶋美雄, 鉄谷信二, 伴野明, 岸野文郎 : “視点追跡を用いた広視域立体視表示技術”, 信学論 (C-II), J75-C-II, 11, pp.719-728 (1992) .
- [74] 片山昭宏, 田中宏一郎, 押野隆弘, 田村秀行 : “多視点画像データの補間処理による視点追従型立体画像表示”, 3次元画像コンファレンス'94, pp.7-12 (1994) .
- [75] S. A. Benton : “ホログラフィックビデオ画像に感ずる実験”, 3D映像, 5, 2, pp.57-68 (1991) .
- [76] 橋本伸幸 : “新技術を用いた動画ホログラフイー”, 画像ラボ, 2, 12, pp.58-62 (1991) .
- [77] 曾山豊, 山内博, 安田孝美, 横井茂樹, 鳥脇純一郎 : “景観シミュレーションのための2.5次元簡易情景モデル構築の一手法”, 画像の認識・理解シンポジウム (MURU'92), pp.439 (II) -446 (II) (1992) .
- [78] P.Metzger : “Adding reality to the virtual”, VRAIS'93, pp.7-13 (1993) .
- [79] P.Milgram : “Mixed reality : Research issues in merging real and virtual worlds”, ATR Workshop on Virtual Space Teleconferencing, 1993, (1993) .
- [80] P.Milgram, S.Zhai, D.Drascic : “Applications of augmented reality for human-robot communication”, Proc.Int.Conf.Intelligent Robots and Systems, pp.26-30 (1993) .
- [81] S.Feiner, B.Macintyre, P.Seligmann : “Knowledge based augmented reality”, C.ACM, 36, 7, pp.53-62 (1993) .

- [82] W. Barfield, T. A. Furness : “Virtual Emvironments and Advanced Interface Design” , OXFORD UNIVERSITY PRESS, (1995) .
- [83] 中嶋正之 : “最先端コンピュータグラフィックス : 3次元CGの手ほどき, CGの基礎論” , テレビ誌 , 46, 1, pp.49-57 (1992) .
- [84] 岡田稔, 横井茂樹, 鳥脇純一郎 : “画素シフト法に基づく非解析的テクスチャとその自然対象物生成への応用” , 情処学研報 , 89-CG-40-2, (1989) .
- [85] 池田光男 : “眼はなにを見ているかー視覚系の情報処理ー” , 平凡社, (1988) .
- [86] 田崎京二, 大山正, 樋渡涓二 : “視覚情報処理” , 朝倉書店, (1979) .
- [87] 樋渡涓二 : “生体情報工学” , コロナ社, (1971) .
- [88] 大山正, 今井省吾, 和気典二 / 編 : “新編 感覚・知覚心理学ハンドブック” , 誠信書房, (1994) .
- [89] 内山晋二, 山本裕之, 田村秀行 : “距離画像からの階層化適応型パッチ生成法” , 情処学論 , 36, 2, pp.351-361 (1995) .
- [90] 千葉則茂 : “樹木・草本の生成に関する研究動向” , グラフィックスとCAD , 40-12, pp.81-86 (1989) .
- [91] 中嶋正之 : “CGによる自然物体の生成技法” , グラフィックスとCAD , 40-10, pp.69-74 (1989) .
- [92] 中嶋正之 : “CGによる自然物体の生成に関する研究動向 (第2報)” , グラフィックスとCAD , 52-1, pp.1-11 (1991) .
- [93] 大野義夫 : “植物を対象とするCG技術の研究動向” , グラフィックスとCAD , 52-3, pp.13-18 (1991) .
- [94] M.Aono, T.L.Kunii : “Botanical Tree Image Generation” , IEEE Computer Graphics & Applications, 4, 5, pp.10-34 (1984) .
- [95] P.Reffye, C.Edelin, J.Francon, M.Jeager, C.Puech : “Plants Models Faithful to Botanical Structure and Development” , Computer Graphics, 22, 4, pp.151-158 (1988) .
- [96] 中嶋正之, 福田智美, 安居院猛 : “景観表示のための樹木の生成方法” , NICOGRAPH論文コンテスト , pp.22-29 (1988) .
- [97] 金丸直義, 斎藤伸自, 千葉則茂, 高橋清明 : “向日性による樹木の自然な枝振りのCGシミュレーション” , 信学論 (D-II) , J75-D-II, 1, pp.76-85 (1992) .

- [98] 千葉則茂, 大川俊一, 村岡一信, 三浦守 : “CGのための樹木の成長モデル—架空の「植物ホルモン」による自然な樹形の生成—”, 信学論 (D-II), J76-DII, 8, pp.1722-1733 (1993).
- [99] G.Lafue : “Recognition of three-dimensional objects from orthographic views”, ACM/SIGGRAPH, pp.103-108 (1976).
- [100] H.Sakurai, D.C.Gossard : “Solid model input through orthographic views”, ACM/SIGGRAPH, pp.243-252 (1983).
- [101] P. Giblin, R. Weiss : “Reconstruction surface from profiles”, 1st ICCV, pp.136-144 (1987).
- [102] R. Cipolla, A. Black : “The dynamic analysis of apparent surface”, 3rd ICCV, pp.616-623 (1990).
- [103] J. Y. Zheng, 岸野文郎 : “連続シルエットを用いた3次元モデルの復元とその未知領域の検出”, 信学論 (D-II), J76-D-II, 6, pp.1114-1122 (1993).
- [104] 安居院猛, 小峯賢治, 長橋宏 : “シルエット画像を用いた鉢植え植物の3次元モデリング”, 1995年信学総全大, 情報・システム2, pp.380 (1995).
- [105] 横矢直和 : “フラクタルによる3次元複雑形状の解析とその応用”, 信学技法, PRU86-23, (1986).
- [106] 荒川賢一, エリック クロトコフ : “フラクタル幾何を適用した自然地形のモデリング”, 信学論 (D-II), J-76-D-II, 12, pp.2564-2577 (1993).
- [107] A.E.Jacquin : “Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations”, IEEE Trans. Image Process., 1, 1, pp.18-30 (1992).
- [108] T. Mista, J. Quian, J. R. Galvin : “3-D Modeling of Lung Morphogenesis using Fractals”, SPIE, 1898, pp.540-548 (1993).
- [109] 本条毅, 斎藤馨, 熊谷洋一 : “植物形状モデリングとその可視化による景観予測に関する考察”, 造園雑誌, 55, 5, pp.301-306 (1992)
- [110] 森本幸裕 : “植物形状モデリング・可視化システムを用いた桂離宮庭園の植生景観のシミュレーション”, 造園雑誌, 57, 2, pp.113-120 (1992)
- [111] 森本幸裕 : “植物形状モデリングによる緑化景観予測システムの応用について”, 日本緑化工学会誌, 18, 3, pp.162-167 (1993)
- [113] 斎藤馨, 熊谷洋一, 本条毅, CHO Tong-Buhm, 吉岡太郎, 筒井一貴 : “GIS, CAD, 植物成長モデルを応用した景観シミュレーション手法に関する研究”, ランドスケープ研究, 58, 5, pp.197-200 (1995)

- [114] B. T. Milne : “The utility of fractal geometry in landscape design” , Landscape and Urban Planning, 21, pp.81-90 (1991) .
- [115] 横井茂樹, 岡田稔 : “ランダム・フラクタルを用いた木目と大理石の表現技術” , PIXEL, 89-CG-40-2, 71, pp.87-92 (1988) .
- [116] 長尾真 (監訳) : “デジタル画像処理” , 近代科学社, (1978) .
- [117] 黒瀬能のぶ, 松島勇雄, 松尾俊彦 (共著) : “C言語による科学技術計算サブルーチンライブラリ” , 啓学出版, (1986) .
- [118] 山口博幸, 伴野明, 岸野文郎 : “視線検出を利用する広視野高精細表示方法の検討” , 信学論 (D-II) , J73-C-II, 11, pp.721-732 (1990) .
- [119] 畑田豊彦, 坂田晴夫, 日下秀夫 : “画面サイズによる方向知覚誘導効果” , テレビ誌, 35, 5, pp.407-413 (1979) .
- [120] A. P. Pentland : “Fractal-based description of natural scene” , IEEE Trans., PAMI-6, 6, pp.661-674 (1984) .
- [121] M. C. Stein : “Fractal Image Models and Object Recognition” , SPIE, 845, pp.293-300 (1987) .
- [122] M. Hild, 白井良明 : “コア領域を用いるシーンの解釈” , MIRU'94 Proceedings, Japan, pp.I-329-I-336 (1994) .



## 関連するソフトウェア

下記のソフトウェア，及びドキュメントがATR 知能映像通信研究所に保管されている。

ソフトウェア名称： 景観画像生成支援ソフトウェア（景観画像生成プログラム）

ドキュメント：	景観画像生成プログラム「tree_exe」	機能操作説明書
	同	システム説明書
	同	プログラム仕様書