

手書きスケッチと例示予測インタフェースに基づく インタラクティブな3次元樹木モデルのデザイン手法

岡部 誠
東京大学

大和田 茂
東京大学
Sony CSL

五十嵐 健夫
東京大学
PRESTO, JST

概要

手書きスケッチと例示予測インタフェースに基づく、3次元樹木モデルを迅速かつ簡単にデザインするためのシステムを提案する。提案システムは、枝間の距離をなるべく大きくするように枝を広げることにより、樹木の2次元スケッチを3次元化する。また、例示予測によって、より多くの枝や葉をデザインすることができる。ユーザテストの結果から、本システムが初心者にとって扱いやすく、様々な3次元樹木モデルをデザインできることが分かった。

1. はじめに

3次元樹木モデルは、都市景観デザイン、VR（仮想現実）、ゲームなど様々な分野で必要とされる。しかし、樹木の構造は複雑なため、モデリングは困難であると言える。現在の3次元樹木のモデリングには主に2つの手法がある。1つにはL-systemsに代表される生成規則に基づく手法である。生成規則による樹木モデルやシミュレーションは非常にリアルなものであるが、生成規則とパラメータ調節は熟達しなければ難しい。もう1つの手法は、既存の3Dモデルをそのまま、もしくはパラメータ調節して使用するものである。しかし、結果の3D樹木モデルが初期モデルに大きく依存し、デザインの自由度が低い。

我々は手書きスケッチと例示予測インタフェースに基づいた3次元樹木モデリングシステムを提案する。手書きスケッチは迅速な3次元モデルの構築を可能にし[10, 26]、例示予測インタフェースは繰り返し操作を自動化する[3, 15]。これらにより、ユーザは直感的にモデリングを行うことができる。

我々は手書きスケッチの3次元化アルゴリズムの設計にあたって、次の2つの仮定をおいた。1つは、樹木は枝を、枝同士の距離がなるべく大きくなるように広げる、という観察結果に基づく仮定であり、もう1つは、樹木をスケッチする際、主に横方向に広がる枝を描き、前後方向に広がる枝は描かない、というユーザの傾向に基づく仮定である。

また、本システムでは「枝の複製」「葉の配置」「伝播」の3種類の例示予測を行うことができる。この予測には、樹木の一般的な形態学的性質を応用している。

図1は手書きスケッチと、これらを基に我々のシステムで生成された3次元樹木モデルの例である。我々の目的は自然の法則をシミュレートすることではなく、手書き入力を通じて、ユーザの意思をより反映させたモデルを短時間でデザインすることにある。現在のプロトタイプシステム

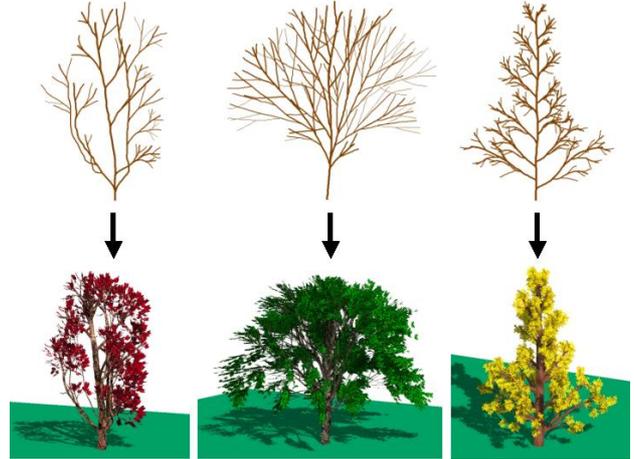


図1. 2次元スケッチ（上段）と最終的に得られた3次元樹木モデル（下段）。左から若木、ケヤキ、イチョウ。提案アルゴリズムにより2次元スケッチから3次元樹木を自動的に作成し、更に枝葉を追加した。各々は数分の処理である。

はターゲットを樹木に絞っているが、提案するインタフェースは他の植物のデザインにも応用が効くものと考えている。

2. 既存研究

2.1. 植物のモデリング

生成規則に基づく手法としては、1968年にLindenmayerが提案し、以後Prusinkiewiczが改良したL-systemsが有名である[11, 19]。最近では、植物と環境の相互作用[16, 20]や、位置情報を用いたインタラクティブなモデリング環境について研究が行われている[21]。他には、初期のL-systemsの限界に言及したAonoらの研究[1]や、Weberらの研究[25]がある。

Deussenらは生成規則とパラメータをGUIで扱えるXfrogを発表した[5]。Deussenらはまた、樹木のNPR手法も発表している[6]。他には写真から樹木を生成する手法も存在する[22, 23]。また、Maierhoferらは生成規則のパラメータを写真から得る、インタラクティブな手法を提案した[14]。また最近ではBoudonらが盆栽生成手法を提案している[2]。

2.2. スケッチベースモデリング

2次元の入力、主に線画から3次元モデルを構築する手法は、制約を解くEgglらの手法[7]や、最適化ベースのLipsonらの手法[12, 13]、最小化問題を解くPentlandらの手法[18]、対称性を利用したTanakaらの手法[24]があるが、この他にも数多く存在する。我々が特に興味をもっているのは2次元ジェスチャを利用した、インタラクティブ性の強いスケッチベース3次元モデラであり、ZelevnikらのSKETCH [26]や、IgarashiらのTeddy [10, 17]などの手法である。

Interactive Design of Botanical Trees Using Freehand Sketches and Example-based Editing

Makoto Okabe (The University of Tokyo)
Shigeru Owada (The University of Tokyo, Sony CSL)
Takeo Igarashi (The University of Tokyo, PRESTO, JST)

<http://www-ui.is.s.u-tokyo.ac.jp/~makoto21>

2.3. 例示予測インタフェース

例示予測インタフェースは、ユーザが例として与えた操作に基づき、繰り返し作業を自動化する[4]。Eagerはユーザの繰り返し操作を発見すると、それらの自動化を促すシステムである[3]。Metamouseシステムは、ユーザが明示的に訓練することで、繰り返し操作を自動化する [15]。五十嵐らは2次元線画の整形システムとしてPegasusを提案した[9]。また3次元に拡張したChateauも提案している[8]。

3. ユーザインタフェース

3.1. 概要

図2に我々のシステムを用いて3次元樹木モデルをデザインする際の、典型的な流れを示す。ユーザはまずマウスかペンを使用して樹木をスケッチする(図2(a))。次に“3D”ボタンを押すと、スケッチが自動的に3次元化される(図2(b))。ユーザはジェスチャ入力により枝を切断、消去したり、また新たな枝をスケッチして追加することができる。また、例示予測を使用することで、より複雑な樹木形状をデザインできる。予測には3種類あり、ボタンにより区別される。「枝の複製」モードでは、指定した親枝の周囲に、既に存在する枝を基に、枝の追加が行われる(図2(c))。「葉の配置」モードでは、形態学的によくある葉の生え方(双生、互生、輪生)をシステムが例示し、ユーザはそれらを選択するだけで、枝の周囲に葉を自動生成することができる(図2(d))。最後に「伝播」モードにより、子枝集合を他の親枝にコピーすることで、全体的に茂った樹木を生成することができる(図2(e))。

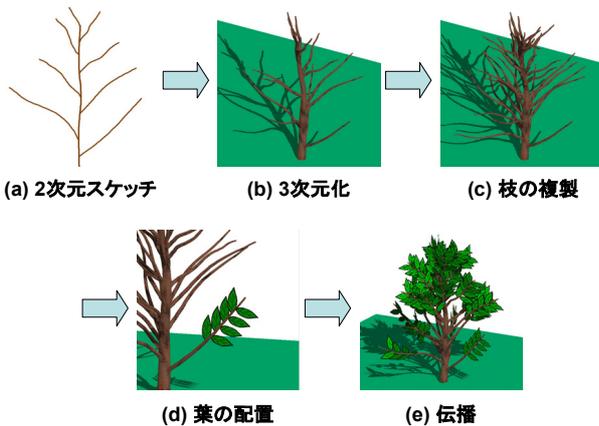


図2. 3次元樹木モデリングの流れ。(a) 樹木の2次元スケッチ、(b) “3D”ボタンが押されシステムが3次元樹木モデルを生成、(c) 複製モードにより枝数を増やす、(d) 葉の配置モードにより葉を追加、(e) 伝播モードにより葉を他の枝に伝播

3.2. 樹木の2次元スケッチ

ユーザは枝や葉を表す2次元ストロークによりスケッチをする。開いたストロークは枝となり、閉じたストロークは葉を表現するポリゴンとなる。入力されたストロークは最も近い枝に接続するように変形される。つまり、ストロークの根元の点は接続位置まで移動し、ストロークの先端の点は全く動かず、他の点は補間により適切な位置に移動される(図3(a,b))。

図3(a,b)のように、ユーザが枝を1本のストロークで描いた場合には、システムはそのストロークに沿って半径が線形に細くなる円板をスweepした枝形状を、3次元化処理の際に生成する。また、1本の枝を2本のストロークで描くことによって、太さの変化を定義することもできる

(図4)。1度太さの変化が定義されれば、次からは1本のストロークを描いた場合でも、その形状が適用される。

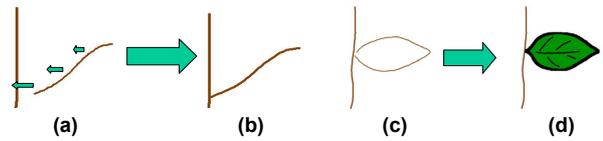


図3. (a-b) 枝のスケッチ (c-d) 葉のスケッチ

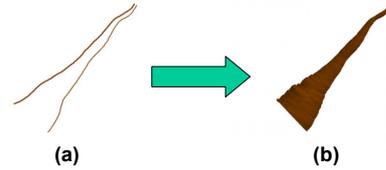


図4. (a) 2本のストロークで枝を描く、(b) 結果の枝の3次元形状

ユーザはまた、ジェスチャ操作により、枝の切断、消去を行うことができる。枝と1回で交わるストロークは枝を切断し、切断点よりも先端側の枝葉が消去される。また枝と2回以上で交わるストロークは枝をつけ根から消去する。

3.3. 2次元スケッチから3次元樹木形状の生成

ユーザが2次元スケッチを描き終わり、“3D”ボタンを押すと、樹木が3次元化される(図5)。このプロセスには数秒を要する。3次元化が終わると、樹木を回転させ、任意の方向から見る事ができる。3次元化アルゴリズムについては、4章で詳しく述べる。

3次元化が完了したら、2次元スケッチの場合と同様に、スケッチを追加することで枝や葉を追加したり、ジェスチャ入力により枝を切断、消去することもできる。

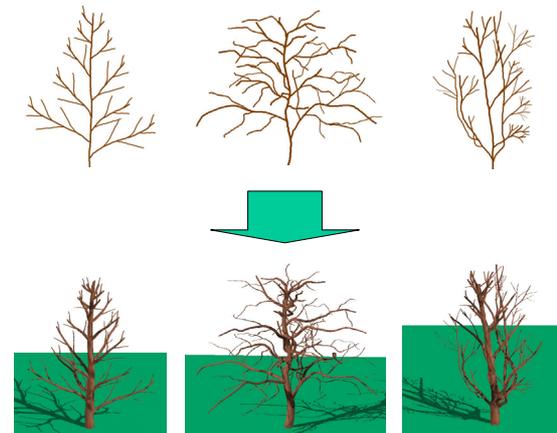


図5. 上段は樹木のスケッチ、下段は3次元化の結果

3.4. 例示予測に基づく枝の複製

この複製モードでは、枝を指定し、更にはその枝を複数回クリック、もしくはマウスを右回転させるジェスチャにより枝を増やすことができる。またマウスを左回転させるジェスチャにより枝を減らすことができる。図6に示すように、枝の増減に関わらず、樹木の全体的な形状は保存される。追加される枝の位置、方向、形状を決定するアルゴリズムは4章で詳しく説明する。

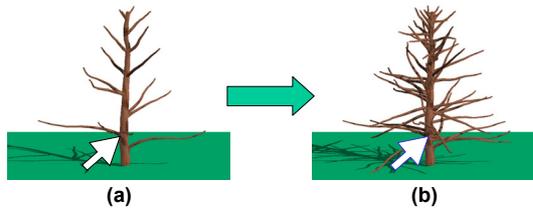


図6. 枝の複製モードによる枝の自動生成。幹の周囲により多くの枝を生成している

3.5. 例示予測に基づく葉の配置

このモードでは、植物形態学的な葉の生え方のパターン、双生、互生、輪生に基づいて葉を生成することができる。ユーザはまずスケッチにより樹木に葉を加える。すると図7に示すように、システムが、それらから考えられ得る葉の生え方パターンをいくつか自動生成し、画面上にサムネイル表示する[8]。ユーザはサムネイルをクリックしてそのパターンを適用することもできるし、結果を無視して編集を続けることもできる。

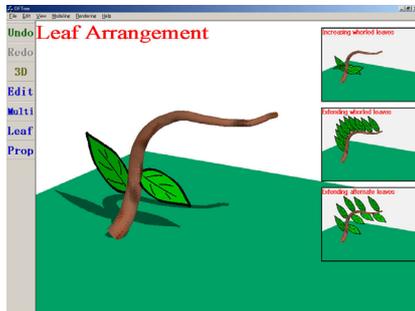


図7. システムは葉の生え方をサムネイルで示す。サムネイルをクリックすることで実際のモデルに適用する

1つ目は輪生の葉集合に新たに葉を追加する例示予測である(図8)。ほぼ同じ位置に葉の集合を見つければ、システムは、ユーザがそこに更に葉を増やしていきたいのではないかと予測する。システムは葉を1枚追加した後、葉集合が輪生となるように再配置する。この例示予測により、表示されるサムネイルを続けてクリックすることによって、輪生の葉の数を増加させることができる。

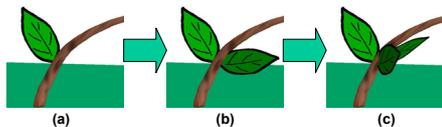


図8. 輪生の葉集合の数を増やす

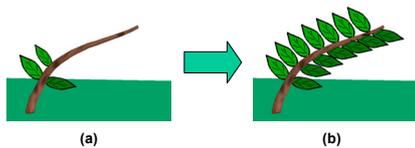


図9. 親枝の周囲に輪生の葉集合を伸ばし、全体として双生葉集合を生成している

2つ目は輪生の葉集合を枝に沿って拡張しようとする例示予測である(図9)。ほぼ同じ位置に葉の集合を見つけるとともに、それより上位の位置に別な葉を見つけると、システムは、ユーザがその枝上に葉の集合を、上位に見つけた葉との間隔で枝上に拡張していきたいのではないかと

と予測する。システムは枝の上位部に葉集合を拡張して、サムネイルを表示する。

3つ目は互生の葉集合を枝に沿って拡張しようとする例示予測である(図10)。単一の葉を異なる位置に複数発見した場合、システムは、ユーザがそこに見出される互生パターンで、枝上に葉を拡張したいのではないかと予測する。この例示予測では、葉を適切に配置しておけば、螺旋構造で葉を配置することが可能である。互生パターンで枝の最上部まで葉を拡張した結果をサムネイルとして表示する。

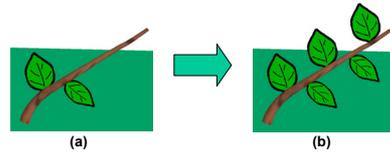


図10. 親枝の周囲に互生の葉を伸ばす

3.6. 例示予測に基づく枝と葉の伝播

このモードでは、ユーザは局所的な枝や葉の生え方を樹木全体に伝播させることができる(図11)。このモードにおいては、枝と葉を区別しない。ここでは、葉を例に説明する。葉を持つ親枝をクリックすると、システムは親枝に生える葉集合を他の全ての枝上に伝播する。現在のシステムでは2つの伝播パターンをサポートしており、それぞれのパターンはボタンで選択する。

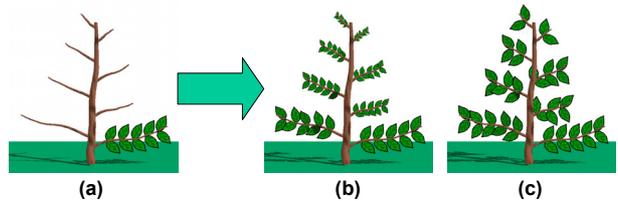


図11. 伝播モード。(b)は個数固定、(c)はサイズ固定で伝播した結果

1つ目の伝播パターンは個数固定(この場合は10枚の葉)でサイズを調整して伝播させるものである。クリックされた親枝が、伝播先の枝にそのサイズをマッチさせるように、葉集合のサイズを調節する。

2つ目の伝播パターンはサイズ固定で個数を調整して伝播させるものである。もし伝播先の枝がクリックされた親枝よりも短かった場合には、システムはクリックされた親枝上で先端に近い部分に生えている葉から順に伝播を行う。

ここでは葉を例にとったが、同様のアルゴリズムは枝の伝播にも用いることができる。個数固定の伝播パターンは、枝の伝播に適し、サイズ固定の伝播は葉の伝播に有用である。これは自然な樹木の場合にも、全体で見れば葉のサイズや生えている間隔というのはほぼ一定である、また枝についてはその逆である、という観察結果に基づく。

4. アルゴリズム

4.1. 2次元スケッチの3次元化

ここで述べるアルゴリズムは、2次元スケッチで描かれた枝の奥行き情報を計算するためのものである(図12)。

計算結果の3次元モデルは、あらゆる角度から見たときに同じような外観になるのが理想である。ある1本の枝は、その枝と他の枝との距離がなるべく大きくなるように、その位置を決定しようとする。この場合、樹木の形状には最適解が存在するはずであるが、今回開発したシステムでは、そのような最適解を求めるのではなく、幹をルートとする木構造だとすると、幹からより若い世代へ順に処理を施し

ていく。ただし、同じ世代間の処理順序はランダムに行う。処理を高速化するため、全ての枝を地面に投影した2次元距離場を使用して最も枝が疎な領域を探し、枝を逐次追加するとともに、距離場を更新する。現在の実装では2次元距離場は128×128のグレイスケール画像として保存されている。

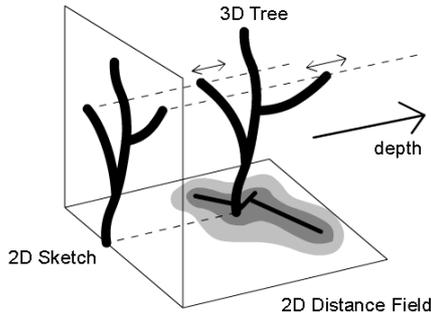


図1 2. 2次元距離場による枝の奥行き情報計算

距離場を用いるだけでは、枝が奥行き方向に過度に伸びてしまう、という現象を押さえ込むことができない。これを防ぐために、2次元スケッチを囲む2次元凸包を得て、それを3次元化した3次元包を計算し(図1 3)、枝位置の検索範囲をこの3次元包の内部だけに絞る。

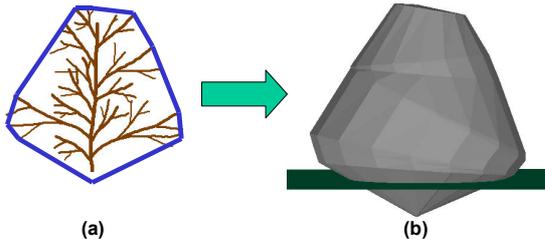


図1 3. 全体的な形状を保存するために作成する2次元凸包を押し出した3次元包

3次元包は2次元凸包の最下点から最上点にかけて円板をスweepすることで作成し、更に、結果の3次元包はルート2倍に拡大したものを用いる。この拡大処理は2次元凸包に接している枝が前後に広がる自由を与えるために必要な処理である。

また、ある枝の長さは、その親枝よりも短い、という観察事実に基づき、枝の長さにも制限を与える[25]。この長さの制限も、2次元から3次元に膨らむためのリラクゼーションとして、1.2の倍率で拡大して用いる。

更に親枝とその子枝が作る角度にも制限を与える。まず2次元スケッチに対して、親枝とその全ての子枝とが作る角度の最大値を求める。その最大値を1.2倍して、その値を3次元になった際にも、その親枝と子枝間で作られる角度の最大値とする。

以上のプロセスを全ての枝に行う。なお以上のプロセスを行うにあたって、枝は全て始点と終点を結ぶ1セグメントから成る線分として扱われる。これは単純化と効率化のためである。

線分から成る2次元スケッチを3次元化した後、オリジナルの2次元スケッチの枝ストロークに、対応する線分をもとに奥行き情報を与える。この奥行き情報の与え方は4.3で述べる。

4.2. 基本アルゴリズムの拡張

樹木の2次元スケッチを描く際、ユーザは横方向に広がる

る枝を描くのみで、前後方向に広がる枝を省略する傾向がある。しかし、上で述べたアルゴリズムは、2次元スケッチを全ての方向に一律に広げようとするため、結果の3次元樹木モデルは、正面の概観は保たれるが、横方向から見た場合の概観が2次元スケッチと大きく違ってしま(図1 4)。

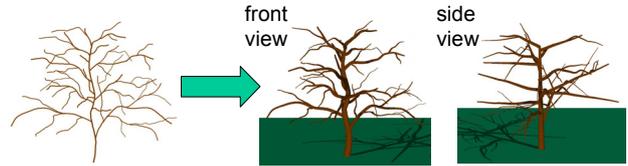


図1 4. 3次元化アルゴリズムの問題点

この問題を解決するために、ユーザは横方向に広がる枝のみを描く、という仮定をおいてアルゴリズムを拡張した。つまり実際に前後方向に伸びる枝はシステムが自動的に追加する。アルゴリズムを拡張するにあたって、4.1で述べたアルゴリズムを適用する範囲は、視線方向のベクトルとなす角度が45度から135度までとする。この範囲で3次元樹木を2つ作成した後、一方を地面から垂直に伸びるベクトルの周囲に90度回転させて、幹を除いてもう一方とマージする(図1 5)。

2つの樹木は、4.1のアルゴリズムのランダム性から少しずつ異なる樹木になる。我々がテストした限りではこの単純なトリックが非常に有効であった。

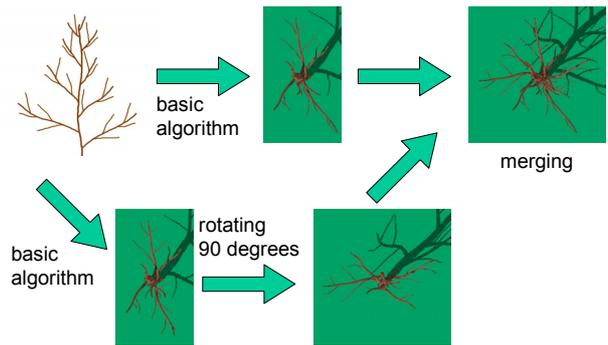


図1 5. 問題解決のためのsimple ad hoc trick

4.3. 枝1本についての奥行き値計算

実際の樹木の枝を見た場合、例えば、波打つような形状をした枝は、横から眺めてみても同様に波打つような形状をしていると考えられる。この効果を与えるために、枝の奥行きを計算するためのアルゴリズムを考案した(図1 6)。このアルゴリズムは、例えばユーザがサインカーブを枝としてスケッチした場合、システムは結果を螺旋形状として出力する。詳細を次に述べる。

元の2次元スケッチは視線に垂直なスクリーン平面上に描かれるので、三次元的には厚みをもたない平らな図形である(図1 6、右上段の式)。z軸方向をストロークの始点から終点へのベクトルとし、視線とz軸に垂直な方向をx軸とする。視線(奥行き)方向はy軸となる。ここで、3次元空間内ではストロークはz軸方向に沿って一定の曲率を持っていると仮定する(図1 6、右下段の式)。この式を解くことで奥行きであるy値を計算することができる。まず一定な曲率を決める。この値はz軸方向の x_i における2階微分の絶対値の最大値とする。この値と、 x_i におけるxの2階微分が得られれば、上式を用いて y_i におけるyの2階微分を計算することができる。

次に、それぞれのyの2階微分値について、その符号を

決定する。最終的に螺旋形状を得るためには、それぞれの符号は、z軸に沿ってxの1階微分値が0となるところで変化する、と考へて決定することができる。

yの y_i における2階微分値の絶対値と符号が決定されれば、積分を2回行うことにより、 y_i の値を計算することができる。yの最初の値を0とし、最後のyの1階微分値が0となるように、yの最初の1階微分の値を調節する。

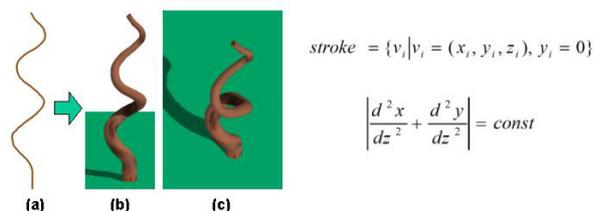


図16. 枝1本についての奥行き値計算

4.4. 例示予測に基づく枝の複製

この操作は、指定された親枝上に既に存在する子枝の情報を元に新たな枝を親枝上に追加する。枝を追加するためには、システムはその枝の位置、長さ、方向、そして形を決定できなければならない。

位置、長さ、形は単純である。システムは新たな枝を、親枝上に生えている枝のペアで、最も離れているものの間に生やそうとする。長さはそれらの枝のペアが持つ長さを補間して決定される。形に関しては同じ親枝に生える枝の中からランダムに1つ選び出してコピーする。

方向は少し複雑である。システムはまず親枝に沿ってL-systemのturtleを走らせ、局所3次元座標系を作成する(図17)。システムは新たな枝の方向を決定するために、2つの角度を用いる。親枝の進行方向(“head vector”)と新たな枝のなす角”down angle”と、進行方向に垂直方向の回転角”rotation angle”である。rotation angleは新たな枝が、進行方向に見て枝が一様に広がって見えるように決定されるが、樹木の枝は下方向には生えない傾向があるので、上方向(up vector)と120度以上の開きをとらないように決定される。down angleは長さと同様に、枝のペアのdown angle情報を補間して決定される。

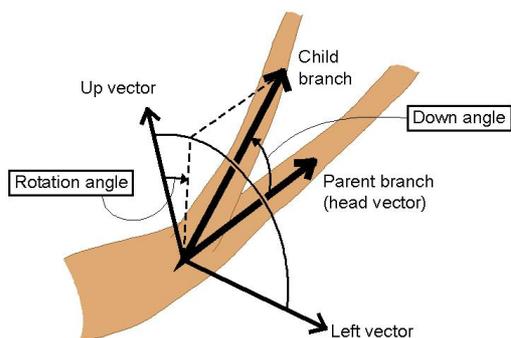


図17. 枝の複製モードで使用する3次元座標系

5. 結果

図18に提案システムで、筆者がモデリングした3次元樹木モデルを掲載する。各々の3次元樹木モデルは平均して10分以内のモデリング時間であった。広葉樹は主に、2次元スケッチを3次元化し、枝と葉の伝播機能を使うことでモデリングできた。また、枝の複製モードは針葉樹をモデリングする際に便利であることが分かった。

提案システムの有用性を調べるために、ユーザテストを

行った。被験者は7名、いずれも提案システムの操作に関しては初心者である。図19は被験者がデザインした3次元樹木モデルと、要した時間である。ここに掲載した樹木モデルのいくつかは、独特ではあるが、自然の樹木と比べると不自然だと思われるものがある。しかし、このような樹木こそユーザの意図をより反映しており、既存の生成規則に基づく手法や、モデリングライブラリを用いる手法などではデザインすることが難しかったものであると考へている。

6. 今後の課題

今回は生成規則などを全く取り入れない形でシステムを設計したが、生成規則は3次元樹木モデルの完成度を高めるために有用である。今後はシステムに生成規則をいかに組み込むか、という問題に取り組みたい。

また今回提案したインターフェースは樹木だけでなく、他の植物にも適用可能なものである。今後は樹木以外の草花などの3次元モデリングにも取り組む。

また現在のシステムは樹木1本を想定している。スケッチの3次元化アルゴリズムにランダム性はあるものの、完成する樹木は多様性に欠ける。今後は1本の樹木から同種と見られる樹木を自動的に生成することにも取り組みたい。

謝辞

ユーザテストに参加して頂いた五十嵐研究室のメンバー、また助言を頂いた東京大学の近山教授、また論文の準備段階で助言を頂いたブラウン大学のHughes教授に感謝致します。また、本研究の成果はIPA(情報処理振興事業協会)「平成15年度 未踏ソフトウェア創造事業」の支援を受けて達成されたものです。

参考文献

1. Aono, M. and Kunii, T. L. Botanical Tree Image Generation. *IEEE Computer Graphics and Applications, Volume 4, No.5, 10-34, 1984.*
2. Boudon, F., Prusinkiewicz, P., Federl, P., Godin, C., and Karwowski, R. *Computer Graphics Forum (Proceedings of EUROGRAPHICS 2003), 22(3), 591-599, 2003.*
3. Cypher, A. Eager: Programming repetitive tasks by example. *Proceedings of CHI, 33-39, 1991.*
4. Cypher, A. Watch What I Do: Programming by Demonstration. Cambridge, MA: MIT Press, 1993.
5. Deussen, O. and Lintermann, B. A Modelling Method and User Interface for Creating Plants. *Proceedings of Graphics Interface 97, 189-197, 1997.*
6. Deussen, O. and Strothotte, T. Computer-Generated Pen-and-Ink Illustration of Trees. In *Proceedings of ACM SIGGRAPH 2000, ACM, 13-18, 2000.*
7. Egli, L., Hsu, C., Elber, G., and Bruderlin, B. Inferring 3D Models from Freehand Sketches and Constraints. *Computer-Aided Design, 29, 2, 101-112, 1997.*
8. Igarashi, T. and Hughes, J. F. A Suggestive Interface for 3D Drawing. *Proceedings of UIST '01, 173-181, 2001.*
9. Igarashi, T., Matsuoka, S., Kawachiya, S., and Tanaka, H. Interactive Beautification: A Technique for Rapid Geometric Design. *Proceedings of UIST '97, 105-114, 1997.*
10. Igarashi, T., Matsuoka, S., and Tanaka, H. Teddy: A Sketching Interface for 3D Freeform Design. In *Proceedings of ACM SIGGRAPH 99, ACM, 409-416, 1999.*

11. Lindenmayer, A. Mathematical Models for Cellular Interactions in Development, I&II. *Journal of Theoretical Biology*, 280-315, 1968.
12. Lipson, H. and Shpitalni, M. Identification of Faces in a 2D Line Drawing Projection of a Wireframe Object. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.18, No.10, 1000-1012, 1996.
13. Lipson, H. and Shpitalni, M. Optimization-Based Reconstruction of a 3D Object From a Single Freehand Line Drawing. *Journal of Computer Aided Design*, Vol.28, No.8, 651-663, 1996.
14. Maierhofer, S. and Tobler, R. F. Creation of Realistic Plants Using Semi-Automatic Parametric Extraction From Photographs. In Technical report 2002-002, VRVis Research Center, 2002.
15. Maulsby, D., Witten, I. H., and Kittlitz, K. A. Metamouse: Specifying Graphical Procedures by Example, Computer Graphics. In *Proceedings of ACM SIGGRAPH 89*, ACM, 127-136, 1989.
16. Měch, R. and Prusinkiewicz, P. Visual Models of Plants Interacting with Their Environment. In *Proceedings of ACM SIGGRAPH 96*, ACM, 397-410, 1996.
17. Owada, S., Nielsen, F., Nakazawa, K., and Igarashi, T. A Sketching Interface for Modeling the Internal Structures of 3D Shapes. In *Proceedings of Smart Graphics 2003*, 49-57, 2003.
18. Pentland, A. and Kuo, J. The Artist at the Interface. Vision Science Technical Report 114, 1989.
19. Prusinkiewicz, P., Hammel, M., Hanan, J., and Měch, R. L-systems: from the theory to visual models of plants. In *Proceedings of the 2nd CSIRO Symposium on Computational Challenges in Life Sciences*. CSIRO Publishing, 1996.
20. Prusinkiewicz, P., James, M., and Měch, R. Synthetic Topiary. In *Proceedings of ACM SIGGRAPH 94*, ACM, 351-358, 1994.
21. Prusinkiewicz, P., Mündermann L., Karwowski, R., and Lane, B. The use of positional information in the modeling of plants. In *Proceedings of ACM SIGGRAPH 2001*, 289-300, 2001.
22. Sakaguchi, T. and Ohya, J. Modeling and Animation of Botanical Trees for Interactive Virtual Environments. In *Proceedings of VRST 99*, 139-146, 1999.
23. Shlyakhter, I., Rozenoer, M., Dorsey, J., and Teller, S. Reconstructing 3D Tree Models from Instrumented Photographs. *IEEE Computer Graphics and Applications*, 21, 3, 53-61, 2001.
24. Tanaka, T., Naito, S., and Takahashi, T. Generalized Symmetry and its Application to 3D Shape Generation. *Visual Computer*, 5, 83-94, 1989.
25. Weber, J. and Penn, J. Creation and Rendering of Realistic Trees. In *Proceedings of ACM SIGGRAPH 95*, 119-128, 1995.
26. Zeleznik, R. C., Herndon, K. P., and Hughes, J. F. SKETCH: An Interface for Sketching 3D Scenes. In *Proceedings of ACM SIGGRAPH 96*, 163-170, 1996.

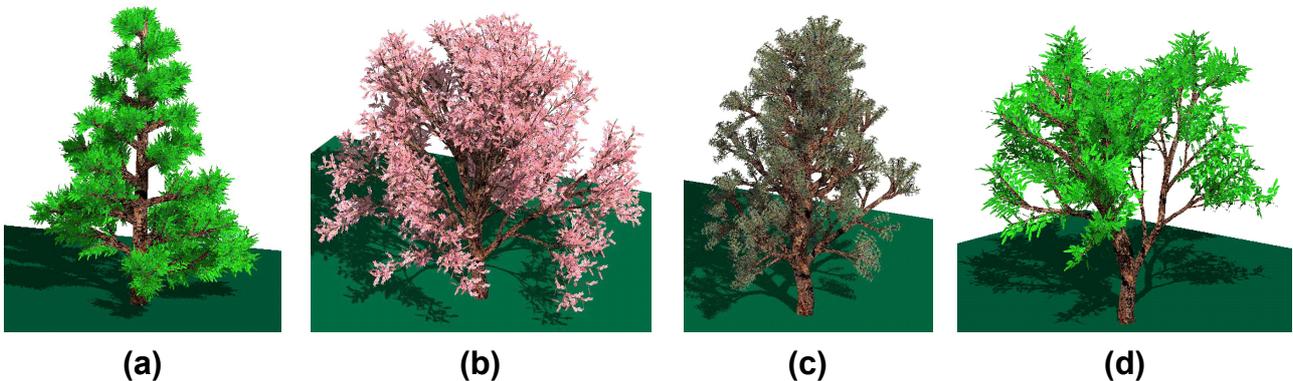


図18. 筆者による3次元樹木モデル。(a) サクラ, (b) マツ, (c) オリエントブナ

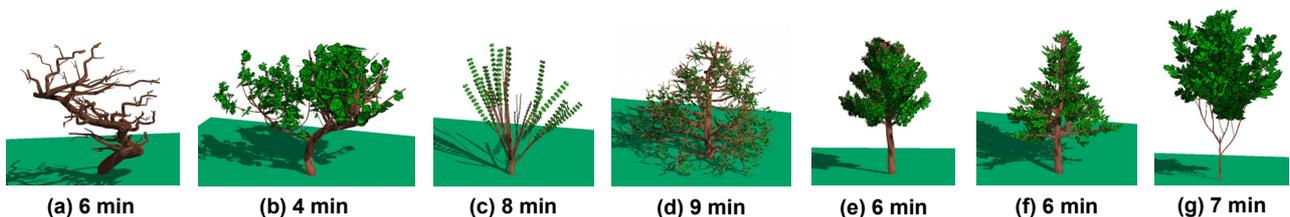


図19. 被験者によるモデリング例