
スケッチによる可動階層構造付き 3次元モデル生成手法

Sketch-based Construction of 3D Models with Hierarchy

五十嵐 健夫 Dennis Cosgrove Randy Pausch*

Summary. This paper presents a sketch-based user interface for constructing 3D models with hierarchical structures. A hierarchical model consists of multiple separate parts connected by joints, and a designer can animate the model by specifying the rotation angle at each joint. In traditional modeling systems, the user has to construct each part separately, and connect them one by one specifying the relative position of each part and the pivot (center of rotation) of each joint. Our technique is an extension of extrusion operation introduced in a sketch-based freeform modeling system called Teddy. Whenever the user extrudes the object, the system creates a new part and set pivot at the basis of the extrusion. As a result, the user can construct hierarchical objects easily without specifying relative position and pivot of a part explicitly.

1 はじめに

ビデオゲームに代表される対話的な3次元グラフィクスや、入門者用のアニメーションシステムにおいて利用される3次元モデルの表現手法として、可動階層構造構造を利用したものがあある。これは、オブジェクト全体のサーフェス形状をボーンなどによって滑らかに変形する手法と異なり、オブジェクトを互いに接続された部品からなるものと表現し、接続点での回転角を変更することでオブジェクトの全体構造を表現するものである。この手法は、ユーザが形状をデザインしやすく、また通常の回転行列計算によって実時間でのレンダリングが可能であるといった利点のために広く利用されている。

しかし、このような可動階層構造をもった3次元モデルの生成を通常モデリングシステム上で行うためには複雑な操作が必要である。例えば、胴体の上に頭をつけるといった場合には、胴体と頭をそれぞれ別々にモデリングした後に、それぞれを適切な位置に移動した後、頭が胴体の子となるような階層構造を埋め込み、最後に頭を回転する時の軸となる点を明示的に指定するといった複雑な操作が必要になる(図1)。

* Takeo Igarashi, 東京大学 / Dennis Cosgrove, Randy Pausch, CMU

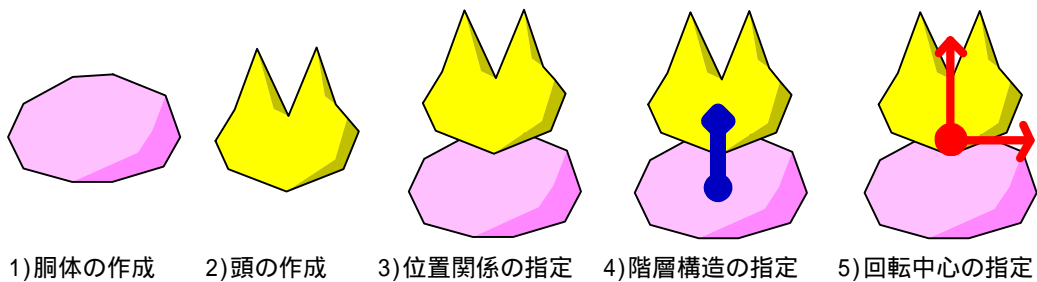


図1: 通常の3次元モデリングシステムでの階層構造の指定操作

本稿では、我々がこれまで開発してきた「初心者でも簡単に3次元モデルを作成できるモデリング手法」[1]を拡張し、可動階層構造をもった3次元モデルを簡単に作成できるようにしたものを紹介する。この手法を利用することにより、ユーザは部品間の位置関係の調整や回転中心の指定などの操作を明示的に行うことなく、ただオブジェクトの形状編集操作を行うだけで可動階層構造を持ったモデルを作成することが可能となる。具体的には、[1]における突起生成操作が行われた時点で、新たに生成された部分を元になるオブジェクトとは別のオブジェクトとして構成し、それらの中に階層構造を指定し回転中心を設定するといった処理を自動的に行う。

2 関連研究

2.1 両手を利用したインタフェース

複数の物体間の位置関係の指定に関しては、3次元入力デバイス等を利用した両手操作インタフェースの有効性が報告されている。特に Pierce ら [5] は、3次元位置センサーを利用したシステムにおいて、左手でベースとなる物体を持ち右手で操作の対象となる物体を持つことで、物体間の相対的位置の指定が効率よく行うことのできる手法を示している。彼らのデモンストレーションでは、兵隊の人形を左手で持ち、右手でその人形のネジを巻く様子が例としてあげられている。

2.2 スケッチによるモデリング手法

3次元モデル生成にかかる手間の軽減を目的として、近年、スケッチを利用したモデリング手法が提案されてきている。これらの手法は、正確なモデリングには適さないものの、ユーザの思い描く形状の概略を簡単な操作によって短時間で構成できるという点で、初心者やデザイン段階での利用に適している。

SKETCH システム[6]は、簡単な2次元のジェスチャーによって幾何形状プリミティブを空間中に配置していくことによって建物や家具などからなる3次元世界

を生成するものである。3次元空間中の位置を2次元の入力から決定する際には、3次元プリミティブは常にすでに存在する他の物体の上に乗っている、という制約を利用している。

Teddy[1]は、それまでのスケッチに基づくモデリングシステムが規則的な形状からなる人工的な物体を対象としていたのに対し、動物や植物といった自由曲面からなる物体を対象としている点を特徴とする。ユーザが画面上に2次元の閉じた線画を描くと、自動的に線で囲まれた領域を膨らませて3次元物体を生成する。また、物体を横切る線によって切断を実現したり、物体表面への描画と表面に垂直な方向への描画を組み合わせることによって突起を生成したりする機能が実装されている。本稿で紹介する手法は、この突起生成機能を拡張するものである。(Teddyは www.mtl.t.u-tokyo.ac.jp/~takeo/teddy/teddy.htm から入手可能)

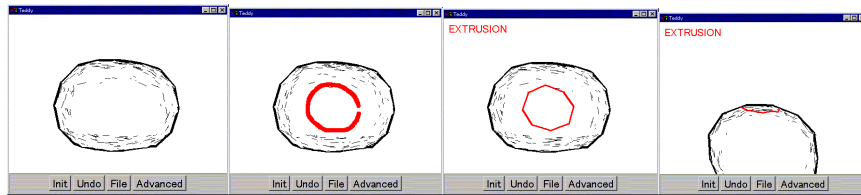
2.3 スクリプトによるアニメーションシステム

階層構造をもったオブジェクトは、キーフレーミングやスクリプトによる手法によって簡単にアニメーションを指定することができる。スクリプトによるシステムの例として Alice [4] が挙げられる。Alice では、`Bunny.head.turn(left)` と記述することによって Bunny オブジェクトの頭が回転する。また `DoInorder` や `DoTogether` といった記述を組み合わせることによって複雑な動きも表現可能である。

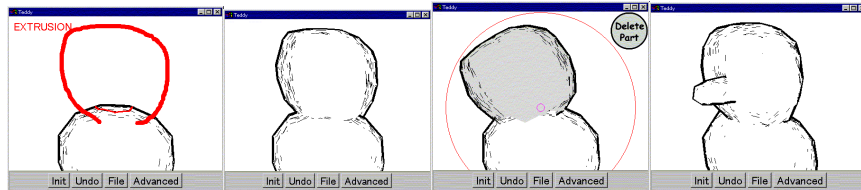
3 スケッチによる可動階層構造付きモデル生成

3.1 ユーザーインタフェース

本稿で紹介する手法は、Teddy[1]における突起生成操作を拡張したものである。ユーザの操作自体は元の Teddy と同様で、まず物体表面に閉じた線を描き(図 7b)、続いて物体を回転(図 7d)、最後に突起の形状を描く(図 7e)というものである。オリジナルの Teddy では、突起が元のモデルと連続したサーフェスとして生成されたが、Teddy2 では突起が元のモデルとは独立したサーフェスとして生成され、元のモデルを親とし新しい突起を子とする階層構造が自動的に割り当てられる。さらに1本目に描かれた閉じた領域の中心が回転中心となる接続点として設定される。ユーザは突起をマウス右ボタンでドラッグすることによって、突起部分をこの接続点周りに回転することができる(図 7g)。突起の上にさらに新しい突起を生成することによって、孫にあたる部分を生成することができる(図 7h)。最終的なモデルは、最初に新規生成操作によって作られた部分を根とする木構造をもったものとして出力される。



a) 初期状態 b) 閉じた線の描画 c) 突起生成モード d) 回転した所



e) 線の描画 f) 突起生成結果 g) 回転した所 h) 「孫」部の生成

図 2 突起生成操作

3.2 アルゴリズム

基本的にはオリジナルの Teddy の突起生成アルゴリズムと同様である。まず 2 本目の線が引かれた時点で、その線を投影するための平面を生成する。この面は物体表面に描かれた 1 本目の線の中心を通りその点における面の法線に垂直な面のうちで、法線が最も視点方向に近いものである。2 本目の線をこの面に投影することで、3 次元の線を得る。次に、この線に沿って 1 本目の線の複製を生成する。最後に隣合う複製の対応する頂点を結んでいくことでメッシュを生成する。根元と先端を綴じあわせることで、突起部を閉じたメッシュとして生成する。元の物体表面のメッシュは一切変更されない。回転中心としては、1 本目の線の中心が利用される。



a) 表面の線 b) 投影面生成 c) 線の投影 d) 表面の線の複製 e) 面生成

図 2 突起生成操作

4 実装

プロトタイプシステムは JAVA プログラムとして実装されている。レンダリングアルゴリズムとしては、[2]のスケッチ風レンダリングアルゴリズムを簡略化したものを実装している。3次元モデルは、階層構造をもつ複数のポリゴンサーフ

エスから構成される(図 3)。各ポリゴンサーフェスは標準的なポリゴンモデルであり、頂点およびエッジ、ポリゴンの集合とその間の連結関係によって表現される。回転操作は部品間の角度を変更するのみであり、個々のメッシュ構造が変形することはない。

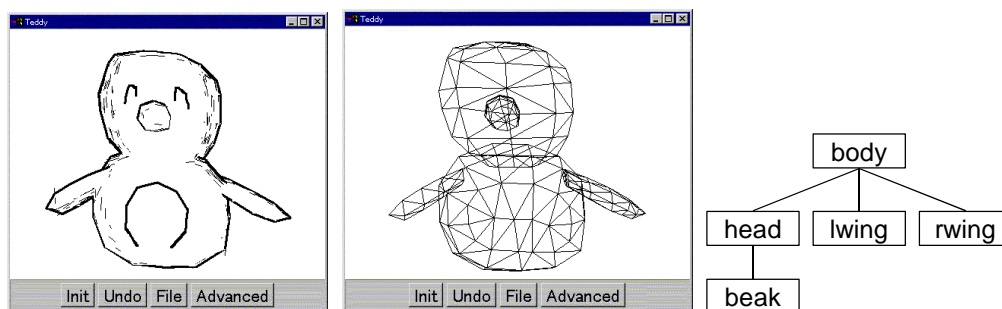


図 3 モデルの内部表現 (左から画面例、メッシュ構造、階層構造)

Teddy2 は Alice から直接呼び出すことが可能となっており、統一された環境の中で、モデルの生成、色付け、アニメーション作成といった一連の作業が可能となっている。(www.alice.orgよりダウンロード可能)。図 4 に例を示す。



図 4 Alice によるアニメーションの例

5 プロトタイプシステムの利用例

1999 年の SIGGRAPH において ALICE システムと Teddy2 を組み合わせたシステムの展示を行い、多くの参加者が階層付きモデルの生成およびアニメーションの生成を体験した。図 5 に参加者によって作成されたモデルの例を示す。チュートリアルを含め、モデリングにかかる時間は数分から数十分以内であった。



図 5 初心者によるモデリング例

6 結論

本稿では、アニメーションに適する階層構造をもった3次元モデルを簡単に生成するための手法およびそのプロトタイプシステムについて紹介した。本手法を利用することにより、3次元CGに馴染みの薄い一般ユーザでも、5節に示したように階層構造をもった多様な3次元モデルを作成することが可能になる。

今後の課題として、複数の物体をつなぎあわせて階層的構造を生成したり、関節での回転に関する制約の指定を行えるようにするといった拡張が挙げられる。また更なる発展としては、現在のような非連続的なサーフェスを用いるのではなく、連続的なサーフェスをもつモデルを利用することによってより滑らかなアニメーションを実現するといった方向性が考えられる。現在、連続的なサーフェスモデルをアニメーションさせる手法としてはサーフェスモデルにボーンを埋め込んで変形する手法が一般的であるが、本稿で紹介した階層的なモデルを骨組みとして、スキン[3]のような手法によって「皮をかぶせる」ことによってより自然なアニメーションが実現できると考えられる。

参考文献

- [1] T.Igarashi, S. Matsuoka, H. Tanaka. Teddy: A Sketching Interface for 3D Freeform Design, SIGGRAPH 99 Conference Proceedings, pp.409-416, 1999.
[www.mtl.t.u-tokyo.ac.jp/~takeo/teddy/teddy.htm]
- [2] L. Markosian, M.A. Kowalski, S.J. Trychin, L.D. Bourdev, D. Goldstein, and J.F. Hughes. Real-time nonphotorealistic rendering. SIGGRAPH 97 Conference Proceedings, pages 415-420, 1997.
- [3] L. Markosian, J.M. Cohen, T. Crulli and J.F. Hughes. Skin: A Constructive Approach to Modeling Free-form Shapes. SIGGRAPH 99, 1999.
- [4] R. Pausch, T. Burnette, A.C. Capeheart, M. Conway, D. Cosgrove, R. DeLine, J. Durbin, R. Gossweiler, S. Koga, and J. White. Alice: Rapid prototyping system for virtual reality. IEEE Computer Graphics and Applications, 15(3): 8-11, May 1995. [www.alice.org]
- [5] J. Pierce, B. Stearns, R. Pausch. Two Handed Manipulation of Voodoo Dolls in Virtual Environments. 1999 Symposium on Interactive 3D Graphics
- [6] R.C. Zeleznik, K.P. Herndon, and J.F. Hughes. SKETCH: An interface for sketching 3D scenes. SIGGRAPH 96 Conference Proceedings, pages 163-170, 1996.