初心者による３次元アニメーションの構築と利用のためのインタフェース

User Interfaces for 3D Computer Graphics

五十嵐　健夫
Takeo IGARASHI

東京大学大学院情報理工学系研究科コンピュータ科学専攻
（〒113-0033 東京都文京区本郷 7-3-1 東京大学理学部 7 号館 303 号室
E-mail: takeo@ui.is.s.u-tokyo.ac.jp ）

ABSTRACT. We developed user interfaces for novice users to design 3D computer graphics and animations quickly and easily. We implemented following four systems. Chameleon system allows the user to paint 3D models. Chateau system.enables the user to design rectilinear models such as houses and castles. SmoothTeddy system allows the user to design 3D freeform models with smooth surfaces. Finally Squirrel system allows the user to quickly design 3D character animation.

１．目的と背景

計算機による３次元コンピュータグラフィクス(CG)は、近年映画やコマーシャルフィルムやビデオゲームに大量に利用されるなど、単にプロダクトとして美しい映像や動画を提示する手法に関しては成熟期を迎えているといえる。一方で、CG のコンテンツ作成は依然として困難な作業であり、技術を習得したエキスパートが膨大な時間をかけて作成しているのが現状である。例えば、３次元アニメーションを生成する際には、各コマにおける３次元モデルの位置や姿勢を基本的にすべて明示的に指定していく必要がある。

本プロジェクトでは、このような状況を改善し、まったくの初心者でも取り扱うことのできるような３次元 CG モデルやアニメーションの構築・利用環境を実現することである。このような子供でも取り扱うことのできるＣＧ制作環境という考え方は、「ＣＧ制作とはそもそも時間と手間のかかるものである」という認識を覆す、日本のみならず世界的に見ても初めての取り組みである(既存の CAD の延長で単純なプリミティブを組み合わせてモデルを作成するような単純なソフトウェアの例はあるが、動物や人間のような自然な形状を表現することは依然として困難である)。

このような技術を実用化することによって、ワープロがすべての人の日常的な道具となり我々の表現の世界を豊かにしたように、３次元ＣＧの応用範囲を格段に広げ、究極的には、計算機を利用したコミュニケーションや情報表現を豊かなものにしていくことが期待できる。

２．成果の概要

本プロジェクトにおいては、以下の４つソフトウェアの開発を行った。

1) ３次元モデルの表面上にテクスチャを塗るシステム (Chameleon)
2)幾何学的な制約を満たす３次元モデルを生成するシステム (Chateau)
3)なめらかな表面をもつ３次元モデルを生成するシステム (SmoothTeddy)
4)空間的キーフレーミングによりアニメーションを生成するシステム (Squirrel)

以下、個々のソフトウェアについて詳細を述べる。

３．３次元モデルの表面上にテクスチャを塗るシステム (Chameleon)

3.1. 概要

３次元アニメーションのための３次元モデル作成にあたっては、幾何形状のモデリングに加えて、表面に色や模様を付加するテクスチャマップの生成が必要である。テクスチャマップ生成にあたっては、特に３次元の物体表面と、２次元のビットマップを結びつける UV マッピング の指定が特に困難であり、ボトルネックとなっている。プロ向けのシステムでは、最適な結果を得るために通常手作業で UV マッピングを指定しているが、初心者向けにおいては、自動で生成できることが望ましい。しかし、既存の自動生成手法では、ユーザのペイント内容にかか

わらず、一方的に固定的な UV マップを設定した上でペイント作業を行うようになっており、不適切な場所に歪みが発生したりビットマップが無駄に使われることが避けられなかった。今回開発したシステムでは、ユーザのペイント操作に応じて、適応的に UV マッピングを生成することで、歪みを最小限に抑え、ビットマップをより有効に利用することを可能としている[3]。また、初心者による手軽なペイント操作を支援する機能として、物体の両面を同時にペイントするレーザーペイント機能や、突起などに隠された部分を塗るときにブラシが自動的に回り込む機能などを実現している。

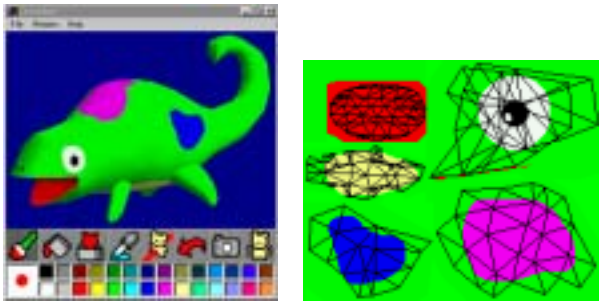


図 1: Chameleon システム
左) 画面例　　右)　自動生成された UV マッピング

3.1. 開発の詳細

1) ユーザの操作に応じて、UV マッピングを動的に割り当てる部分の実装

この部分は、ペイントによって影響を受けるポリゴンを特定して切り出し、それらに新しい UV 値を割り当てる操作と、ペイントされた内容を新しいビットマップ領域に反映させる操作からなる。ペイントによって影響を受ける部分の特定は、ブラシストロークを構成する各線分をポリゴンモデル表面に投影してそれらに交わるポリゴンを見つけることで実行される。また、それらに割り当てられる UV 値は、ペイントされたポリゴンの画面上における座標値を利用する。その際、ペイントされたポリゴン全体を含む bounding box を基準とする。ペイントされた内容をビットマップに反映させるためには、新しく割り当てたビットマップに、まず下絵としてブラシに触れたポリゴンの元のテクスチャ内容を投影して描いた上に、ユーザのストロークを通常の 2 次元のブラシストロークとして描く。

2) ペイントがすべて終わった後に、テクスチャ片をまとめて一つのビットマップにする部分の実装

この操作においては、さまざまな形状をしたテクスチャ片をうまくまとめて正方形のビットマップにはめ込むパッキングが中心となる。任意の 2 次元形状を最小限の正方形内に配置する問題は、NP 完全であり、最適解を見つけるのは困難である。本実装では、まず、各テクスチャ片を長方形の bounding box で囲み、それらを縦にしたあと、高さ順にならべ、折り返しながら畳み込むというヒューリスティクスを用いている。

3) ペイント中、適切なフィードバックを提示するための工夫

本システムでは、ユーザのブラシストロークが終了したあとで、テクスチャ生成と UV マッピング割り当てを行うので、ペイント中のブラシストロークは 3 次元物体上のテクスチャとしてでなく、画面上の 2 次元ストロークとして表示される。しかし、この方法では、 3 次元物体上に描いた場合に得られるライティングの影響や、手前にある障害物に隠れるといった現象が正確に反映されない。この問題を解消するために、ペイント操作前に、まず一時的な UV マッピングと一時的なテクスチャビットマップを構成して割り当てるように実装を変更した。これは、物体全体を画面に対して投影したような UV マッピングとビットマップを設定し、ユーザの入力をそのビットマップ上に描くことで、入力ストロークに 3 次元的なエフェクトを与えるものである。

4) ユーザの操作を支援する諸機能の実装

物体の両面を同時にペイントするレーザーペイント機能は、ブラシストロークに影響を受けるポリンゴンとして、手前にあるポリゴンだけでなく、裏側にあるポリゴンも含むことで実現される。また、裏表に同じ UV を割り当てることで、同一のビットマップを共有することを可能としている。ブラシの回り込み機能は、ブラシストロークをオブジェクト表面に投影するときに、線分ひとつひとつをばらばらに投影するのでなく、端から順番に連続したポリゴン上へ投影するように計算を行うことで実現されている。また、Perlin のノイズ関数を利用してペイントしたモデルがゆらゆらゆれる機能も実現した。

5) ユーザテスト

被験者に対して、簡単なチュートリアルを行い、その後自由にペイント作業を行ってもらう簡単なテストを実施した。操作自体は、画面上の 3 次元物体を回しながら、ブラシで色を塗るという、ユーザの側からはあたりまえの操作であり、違和感なく使用できることを確認した。既存のシステムの経験者は、自動で UV マップが生成されることについて、感銘を受けていた。個別の機能では、レーザペイント機能が好評であったが、領域内を塗りつぶすフラッドフィル機能に問題が認められた。フラッドフィルについては、本手法のもつ根本的な限界のひとつであり解決は容易ではなく、完全な実装は将来の課題である。

４．幾何学的な制約を満たす３次元モデルを生成するシステム（Chateau）

4.1. 概要

本システム[4,5]**は、開発者が過去に開発した図形の対話的整形システム（**Pegasus[1]**）を拡張し、対称性や平行・直角といった基本的な幾何学的関係を満たす３次元モデルの生成を簡単に行えるようにするものである。具体的には、ユーザが指示した図形要素を元に、システムが適切な操作を予測し結果をサムネイルの形で提示することで描画が進行する。ユーザは、気に入った操作内容が提示された場合には、それをクリックすることで処理を実行することができ、気に入ったものが無かった場合には、無視して作業を進めることができる。**

システムはモジュール構造として実装されており、ユーザは簡単な Java コードを記述することでカスタマイズされた機能を追加することができる。

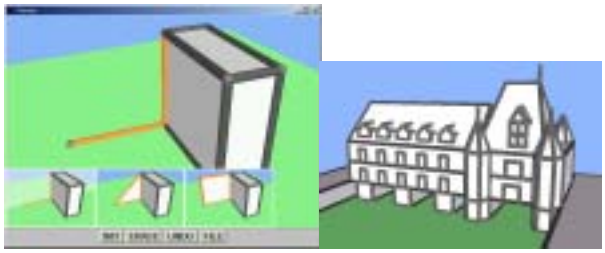


図 2: Chateau の画面例
左）赤でハイライトされた線分をもとに、下部にある３つの候補が生成されている。
右）Chateau によるモデリング例

4.2. 機能の説明

1) 基本操作

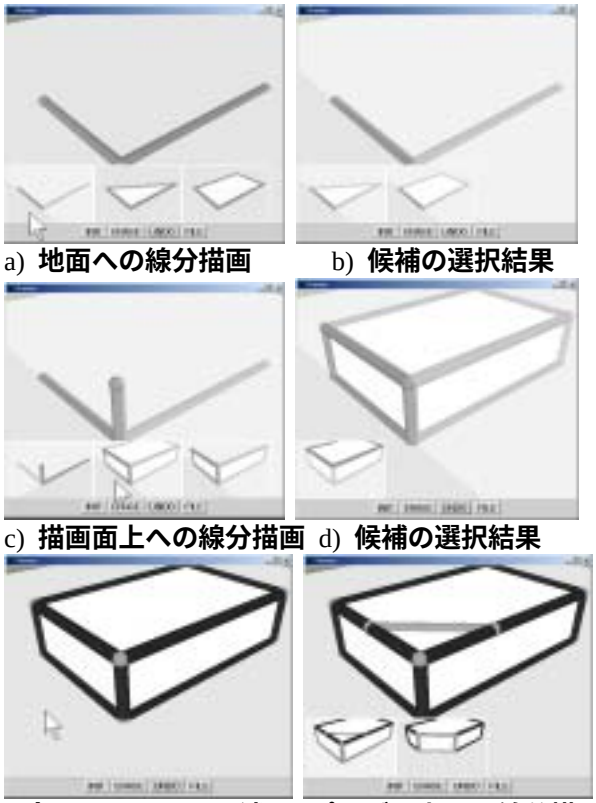
描画操作はすべてマウスの左ボタンで行われ，右ボタンはカメラ操作専用に使われる．左ボタンでのドラッグ操作によって画面上に線が引かれる．描かれた２次元の線分は，画面上で対応する３次元要素の上に投影されて３次元の線分に変換される．具体的には，白い板で表現されたポリゴン，緑色の地面，および後で紹介する「描画面」の上に線を描くことができる．またドラッグ中にマウスを前後に繰り返し動かすことで線分を消去することができる．画面上の既存の３次元線分をクリックすることによって，当該線分をハイライトしたりハイライトを取り消したりすることができる．ダブルクリックすると，クリックされた線分につながっているすべての線分がハイライトされる．ポリゴンをクリックするとそのポリンゴンを囲む全線分がハイライトされ，また何もない部分をクリックするとすべてのハイライトが取り消される．なお，新たに描かれた線分は自動的にハイライトされる．ハイライト操作は，後で説明する線分予測と縮小図による候補提示の他，線分描画時におけるスナップ動作の制御にも使われる．すなわち，ある既存線分がハイライトされていると，描画中の線分はその線分と平行あるいは合同な形になるようにスナップする．スナップ動作については，ドラッグ中に他の要素に触れることでスナップ対象が動的に変化する Drafting Assistant に準じた機能も実装している．

2) 線分の描画予測

本予測機能は，Pegasus システムに実装されていたものとほぼ同等であるが，画面内のすべての線分を走査して適合するものを探すのでなく，ハイライトされた線分のみを対象として予測の手がかりとする．すなわち，新しい線分がハイライトされると，その線分と合同な線分をハイライトされた線分の中から探し出し，その周囲の線分を新しい線分の周りにコピーしてくる．予測結果はピンク色の線分として３次元画面内に直接提示される．気に入ったものがある場合には，クリックすることによって確定操作となるが，無視して次の操作に進むこともできる．本手法は，局所的に対称あるいは合同な形状を描くのに適している．

3) 縮小図による候補提示

ユーザの線分描画・消去操作，あるいはハイライトおよびその取り消し操作が起こると，システムは現在のハイライトの構成に応じて次の操作内容を推定し，縮小図として画面下部に並べて提示する．ユーザは気にいったものがあればそれをクリックすることでメイン画面中の３次元図形へ反映させることができる．気に入ったものがなければ無視して次の操作に進むことができる．
図 3 に操作例を示す．まず，ユーザが画面上に２本線を引くと，それに応じて図 3a のような３候補が提示される．ユーザが一番左側の縮小図をクリックすると，それによって図 3b のような半透明の描画面が表示される．描画面は，既存の図形要素上への描画だけでは実現できない，空中への線分描画に利用される．ユーザがこの描画面上に線分を引くと，結果的に３本の線がハイライトされることになり，それに基づいて図 3c のような３候補が提示される．ユーザが直方体を選んでクリックすると，図 3d のような結果が得られる．ここで地面をクリックすることで，すべての線分のハイライトが取り消され，かつ描画面も消滅する(図 3e)．最後に，直方体の上に線分を引くとそれに応じて，図 3f のように切り落としを含む候補が提示される．



a) 地面への線分描画　b) 候補の選択結果
c) 描画面上への線分描画　d) 候補の選択結果
e)全ハイライトの取消　f)ポリゴン上への線分描画
図 3. 縮小図による候補生成を利用した操作例

各候補の生成は，対応する候補生成エンジンによって生成される．システムには複数の候補生成エンジンが実装されており，各エンジンはそれぞれ，現在のハイライトされた線分の構成と固有の入力パターンを照合し，適合した時に処理を実行して更新された３次元図形を縮小図にしてシステムに返す（図 4）．

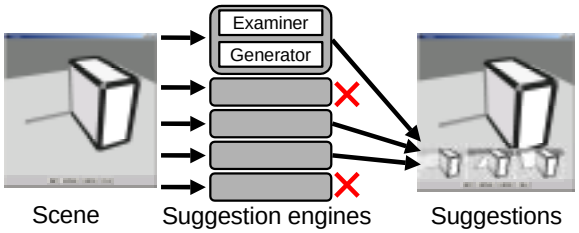


図 4. 候補生成エンジンの動作．入力パターンに適合した場合に結果を返す．

５．なめらかな表面をもつ３次元モデルを生成するシステム（SmoothTeddy）

5.1. 概要

開発者が以前に開発した、手書きスケッチによる３次元モデルシステム（Teddy[2]）は、インターフェース的には優れていたものの、生成されるモデルは、雑なポリンゴンモデルであり、手書き風レンダリングでない通常のレンダリングを行う時や、出来上がったモデルを他のソフト中で使うときには不都合であった。本システムでは、このような問題を解決するために、滑らかな表面形状と整ったポリンゴン形状を生成する手法を開発･実装した。また、滑らかな形状を制御するためのインタフェースの拡張も行った。

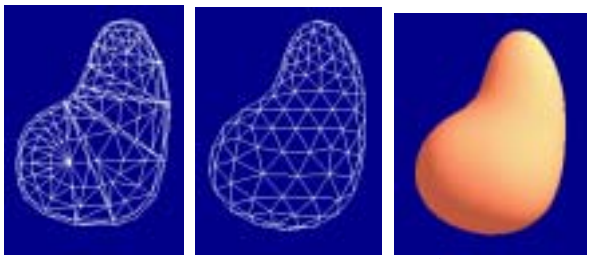


図 5: SmoothTeddy　の画面例
左）整形前のメッシュ　中央）整形されたメッシュ　右）再分割後のサーフェス

図 6: SmoothTeddy　によるモデリング例

5.2. 開発の詳細

1) DirectX を利用したプログラムへの移行

元の Teddy は、Java の AWT のみを使用してレンダリングを行っていたが、表現能力や速度的に限界があるために、３次元 API として DirectX を使用したプログラムに移行した。その際、DirectX の上で、java からシーングラフを扱うことのできる Jalice API を利用した。

2) ポリンゴンメッシュを整えるための SKIN アルゴリズムの実装

SKIN アルゴリズム[6]は、不規則なポリゴンメッシュを整えるための手法であり、頂点を回りの頂点の中心へ動かすステップと、正三角形に近いポリゴンを増やすためにメッシュ構造を編集するステップを交互に作用させる。オリジナルの SKIN アルゴリズムでは、各頂点は参照とするスケルトンメッシュの周りを一定の距離をもって囲むようなサーフェス上を移動するが、本システムでは、以下に述べるようなスケルトンメッシュを補間するようなサーフェス上を移動する。

1. ２次陰関数による滑らかな形状の表現

手書き入力から得たラフな３次元形状から滑らかな表面の表現を得る手法として、ポリンゴン毎に２次陰関数を割り当てる方法を開発した。これは、
Axx+Byy+Czz+Dxy+Eyz+Fzx+Gx+Hy+Iz+J=0　で表現される陰関数を Least Square Fit の手法で計算するもので、この陰関数表現から、局所的な曲率、滑らかな曲面、およびポリンゴンのスムージングを統一的に得ることが可能となっている。

2. 各種モデリング手法の開発

オリジナルのテディではメッシュが粗かったために必要なかった、表面形状を細かく制御するモデリング操作を新たに実装した。具体的には、２つの独立したプリミティブをブーリアン演算によって結合し、さらに接続部を滑らかにつなぐ操作、手書きの曲線によって切断された切り口を滑らかにする操作、表面に引いた線にそって皺を刻む操作などを実装した。また、階層的な部品（胴体に対する腕や足など）を左右対称に生成する操作や、部品を表面上でドラッグして位置の調整を行う操作なども実装した。

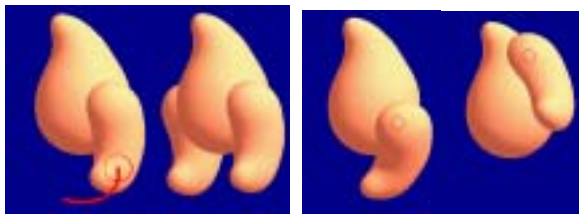


図 7:　SmoothTeddy で新たに実装されたインタフェース例
左）左右対称部品の生成　　右）部品の表面上でのドラッグ

６．空間的キーフレーミングによりアニメーショ

ンを生成するシステム（Squirrel）

6.1. 概要

現在利用されているアニメーション生成手法としては、時間的に離散的な場面における３D キャラクタの姿勢を手作業で細かく指定し、その間を計算機が補間するキーフレーム法がほとんどを占めている。キーフレーム法以外には、体中にセンサーを取り付けて人間の動作を直接取り込むモーションキャプチャと呼ばれる手法もあるが、高価な設備を必要とし個人で使用するものではない。近年、仮想的な３次元世界の中で物理シミュレーションを実行することで自然な動きを構成する手法が活発に研究されているが、これまでの所、プロダクションレベルで高度なアニメーションを生成することに主眼がおかれている。本サブプロジェクトでは、このような既存のアニメーション生成手法に代わり、初心者でも手軽に利用できる手法の開発を行った。具体的には、ユーザはまず、３D キャラクタの姿勢と、３次元空間中のハンドルの位置を結びつける「空間的キーフレーム」を設定する。あとは、ハンドルの位置をインタラクティブに変化させることで、複雑なアニメーションを手軽に生成することができるようになる。

6.2. 開発の詳細

1)　　2 次元上でのテスト

まず、空間的キーフレーム法の基本的なアイデアを確認するために、２次元での実装を行った。本テストプログラムでは、しっぽの形状が、２次元空間中の各点の位置に関連づけられ、マウスをドラッグすることで、しっぽの形状を変化させることができる。

2)　　3 次元上での実装

３次元版では、SmoothTeddy で生成した、階層的な３次元モデルを読み込み、ポーズを設定することができる。画面上には、ピンク色のハンドルが存在しており、set ボタンを押すことで、ハンドルのある場所にキーを打つことができる。キーの設定された場所は黄色いボールで示される。いくつかのキーを３次元空間中に設定後、ハンドルをドラッグすることで、キャラクタのポーズが連続的に変化する。ハンドルの動きを記録して再生することでアニメーションが生成できる。各位置におけるポーズの計算は、設定されたキーにおけるポーズを RadialBasisFunction を用いて補間することで実現されている。具体的な例として、ジャグリング・キック・ダンス・尻文字などの例を作成して有効性を確認した。

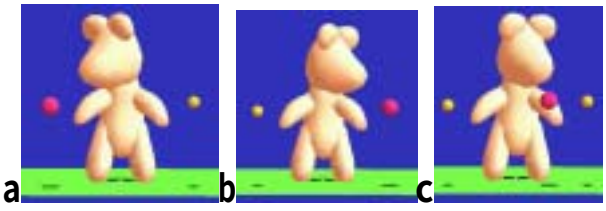


図 9: 2 つのキーを設定した場合の例
左の点が図 a のポーズに、右の点が図 b のポーズに関連付けられている。設定後、ピンクのハンドルを左から右へドラッグすると図 c のように中間のポーズが生成される。ピンクのハンドルは、画面に平行な平面上を移動するが、地面の影をドラッグすることで地面に平行にも移動できる。より多くのキーが設定されている場合にも同様に周囲のキーを元に適切なポーズを生成する。

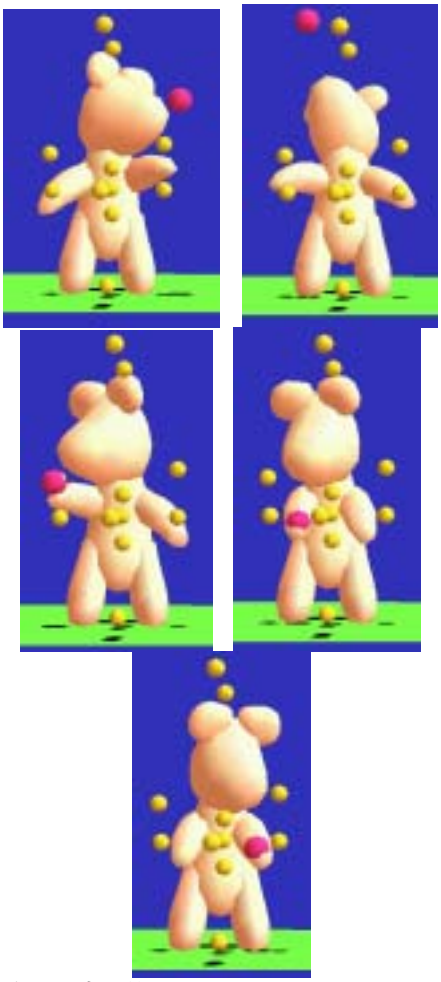


図 10: 3 次元の空間的キーフレームによるジャグリング
黄色い点それぞれに固有のポーズが関連付けられている。システムは、ピンク色のハンドルの位置に応じて周囲の黄色い点に関連付けられているポーズを補間して適切なポーズを生成する。ユーザはピンクのボールをドラッグしていくことで、アニメーションを生成することができる。

3)　　歩行操作の実現

歩行操作のように、キャラクタ自身が移動する場合には、ハンドルを空間中に固定して本体の位置のみを変化させることで、ハンドルのキャラクタの中心に対する相対的な位置を変化させて適切なアニメーションを生成することができる。

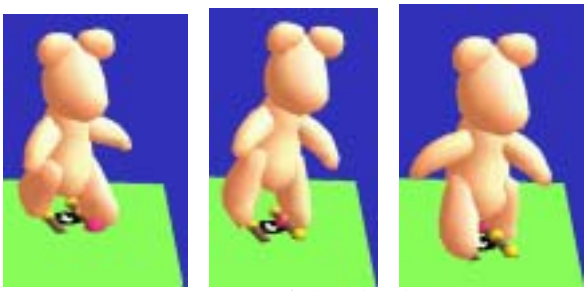


図 11: 歩行動作の実現例
足先の場所に応じて、４つのキーが設定されている。ピ

ンク色のハンドルを地面に対して固定し、キャラクタの位置を移動させることで、ハンドルの相対的な位置が変化し、ポーズがそれに準じて変化している。

７．まとめ

本プロジェクトにおいては、手軽に３次元モデルやアニメーションを生成するためのソフトウェアの開発を行った。個々のソフトウェアは、単体でも十分に利用価値の高いものとなっているが、より本質的な成果は、それぞれの開発過程で得られた、アルゴリズムやインタフェースのデザインにあると考えられる。今後は、これらの成果をもとに、より統合されたシステムを開発していく予定である。

なお、Chameleon, Chateau については、開発者のホームページよりダウンロード可能となっている。
http://www.mtl.t.u-tokyo.ac.jp/~takeo/Welcome-j.html
SmoothTeddy, Squirrel に関しても､順次公開予定である。

Chameleon については、東京大学の技術移転機関を通じて特許申請中であり事業化への営業活動をおこなっている。SmoothTeddy については、エクスツールス株式会社との間で、製品化に向けた検討を行っている。

[1] Takeo Igarashi, Satoshi Matsuoka, Sachiko Kawachiya, Hidehiko Tanaka, "Interactive Beautification: A Technique for Rapid Geometric Design", ACM Annual Symposium on User Interface Software and Technology, UIST'97, pp.105-114, 1997.

[2] Takeo Igarashi, Satoshi Matsuoka, Hidehiko Tanaka, "Teddy: A Sketching Interface for 3D Freeform Design", ACM SIGGRAPH'99, pp.409-416, 1999. (IMPACT PAPER)

[3] Takeo Igarashi, Dennis Cosgrove, "Adaptive Unwrapping for Interactive Texture Painting", 2001 ACM Symposium on Interactive 3D Graphics, pp.209-216, 2001.

[4] Takeo Igarashi, John F. Hughes, "A Suggestive Interface for 3D Drawing", ACM Annual Symposium on User Interface Software and Technology, UIST'01, pp.173-181, 2001.

[5] 五十嵐 健夫, John F. Hughes, "要素指定と候補提示による３次元図形の描画 ", インタラクティブシステムとソフトウェアに関するワークショップ IX (日本ソフトウェア科学会 WISS 2001), 2001 年

[6] Markosian, L., Cohen, J.M., Crulli, T., and Hughes, J.F. "Skin: A Constructive Approach to Modeling Free-Form Shapes", ACM SIGGRAPH '99, pp.393-400.