グラフィクスとCAD 52−13
（1991. 8. 17）

有機的形状を考慮したモデリング及び質感作成に関して

加藤俊明　亀田慶一
ポリゴン・ピクチュアズ
ビッグバンプロジェクト

コンピュータグラフィクスによるキャラクターアニメーション映像を制作する場合には、有機的形状をうまくコントロールする必要がある。ここでは、我々がエンターテイメント映像を制作するために、制作現場から発想した、形状のコントロール手法および、質感作成についての1手法を紹介する。

Consideration on Organic Shape Generation and Attribute Design

Toshiaki Katoh　Keiichi Kameda
Polygon Pictures Inc.
Big Bang Project

Bond Street T11
2-2-43 Higashi Shinagawa, Shinagawa-ku,
Tokyo 140 Japan

It is very important to control organic shape skill fully, when you make character animation with computer graphics. We would like to show you a method how to control organic shape and attribute design, we found out for the making entertainment pictures.

1 目標

我々が目標とする映像は、コンピュータの利点をいかした、コンピュータグラフィクスでしか制作できない非常に手の込んだ映像である。

この最終的な映像のために突破していかなければならない色々な技術的問題を解決するために、現在取り上げている達成目標は、長編のキャラクターアニメーション映像の制作である。キャラクターアニメーション映像と一言で表現しても、技術的には、内容によってかなりの差が生じるが、ここで達成したい技術的項目としては、有機的形状のコントロールに関する諸問題の解決と、長編映像を制作する上での作業工程の確立である。後者においては、プロダクションワーク的ノウハウの部分であるので、本発表では省略する。

2 開発の背景

我々は現在の開発を１９８８年の冬から開始した。現在までのところ、いくつかの技術的トピックスを構築しているが、開発の節目において、そこで新たに完成した手法や、道具の実用性を確かめる意味で短い実験映像を作成している。

3 問題点

コンピュータグラフィクスによるキャラクターアニメーションを考える場合の問題点はなにかを考えてみたい。

剛体として表現された形状と、その形状に関する変換マトリクスで表現する映像は、ロボットや、幾何学的形状の物体の表現には適するが、有機的形状の表現については問題が多い。道具が持っている特性、つまり技術力が作品の表現力と表裏一体の感のあるＣＧでは、現在まで、あまりに多数の剛体物体の映像が制作されてきている。しかしながら、今日このような技術的限界を越えての表現要求が高まり、有機的形状の簡略化され、かつ表現能力の高いコントロールに関する考察が必要と考えられる。

現在もっとも一般的に考えられているポリゴンデータによる自由曲面の近似手法は、ある程度満足のいく緻密な形状を表現しようとした場合のデータ容量は莫大なものとなり、それらのデータを総合的にかつ局所的にうまくコントロールすることがかなりむずかしい。また、最終的に計算される画像の分解度に形状データ精度が足りているかどうかという問題に関しては、細分化もしくは、間引きなどのテクニックが必要であると考えられる。

現在我々は、ポリゴンデータによる形状表現の手法を選択しているが，この最大の理由は、あくまでも映像としての表現が自由であり、説得力のある形が直接的に作成できるということであった。その意味で、ポリゴンデータによる曲面近似手法を根底において、そのデータをうまく変形コントロールする手法を確立する必要性が大きくなった。メタボールや、パッチ曲面によるアプローチは、最終的にポリゴンデータに変換することによって、上位の造形プロセスで使用していくという方針である。ここで問題になるのは、画像の分解度に依存したデータ精度の保持ということであるが、現在は、細分化［1］及びマシンパワーに依存する形でのオーバークオリティーなデータ精度をもつオブジェクトの作成で対応している。

4 環境

ポリゴンデータベースで、キャラクターを作成することを基本に考えた場合の作業環境に関してはどのようなセッティングが考えられるであろうか。

まず莫大なデータを取り扱わなければならない。キャラクター体で３万から６万ポリゴン程度のデータを必要とし、このキャラクタを作成する途中段階で展開される中間データは、１０万から１５万ポリゴン程度はあると想定される。この値は、今後最終的画像のレゾリューションがあがると数倍は増加するものと予想される。これらのデータを視覚的に確認しながら対話的に作業を行って行くためには、もはやグラフィックエンジン無しでは、道具作りが不可能となってきている。

また、コンピュータに行わせる作業のマシンごとの割り振りも重要なポイントとなる。グラフィクスワークステーション上で巨大な編集ツールを動作中にバックヅラウンドでレンダラを動かすようなことは、双方のアプリケーションにとって得策とは言えない。レンダリングに関しては、一般的にどんなマシン上であっても動作させることが可能であるので、コストパフォーマンスの良いＲ

ＩＳＣワークステーション上で、計算させることが一般的である。今日ではアニメーションチェックのためのレンダリングまでは、グラフィクスエンジンを利用することによって、高速に処理することが可能で、実際の高品質のレンダリングに関しては、リモート処理として、ネットワーク上にある高性能ＲＩＳＣワークステーション上でフレーム単位でパラレルに行うことが可能となっていて、最終的に現実的問題として、コストパフォーマンスを向上させることが可能だ。

一般的にいって、必要な時に必要なだけの計算パワーを作業環境内にリーズナブルなコストで増設することができれば良いわけである。莫大な計算量をこなさなければならないレンダリングの処理に関しては、現時点ですでにこの環境を実現できるところまできている。作品ごとに計算量に応じてコンピュータ資源を増設することに関してコスト的な問題はなんとかなる時代が到来した。残念ながら、アニメータが試行錯誤するグラフィクスワークステーションに関しては現時点では動的な資源の追加はむずかしいと考えざるをえない。

5 問題点の分析

キャラクタアニメーションの制作工程のうちアニメーションに関する諸問題を除いた部分で必要になる処理内容をおおざっぱに分類すると次のようになる。実際の作業の順番は、これらの内容を必要に応じて組みかえて進行する。

5.1 形状入力

ゼロからのコンピュータ内部での形状の造形に関しては、幾何学的形状などのある特殊なものに関しては有効であるが、一般的にまだ研究の余地がかなり存在すると考えられる。この問題のむずかしさは、最終的に立体形状の２Ｄのモニターによる３次元的把握の難しさ等も起因し、現時点としては、直接的に高精度の形状を入力する手段の確立が先行して必要である。

クレイモデルを実際に作成することは、形状の品質の向上の上で、チーム全体の共通の認識を深める意味でとても意味のあることだと考えられる．したがってそこで作成されたクレイモデルをダイレクトに入力する手法の確立は、今後コンピュータによる造形手法の進化が大幅に進んでも重要な技法として存在していくことが予想できる。

また，一旦入力した形状を必要に応じて修正を加えるために修正の為のコマンドが必要である．

5.2 形状変形

すでになんらかのアプローチにより作成された形状データをコンピュータ内部で変形するための新しい手法の確立が必要である。直接的に変形処理は、アニメーションの際に必要になるテクニックであるが、結果的には、造形段階で膨大な数のポリゴンデータ頂点を総合的にコントロールする手段としても有効である。人間が最初に頭の中で考える形状は、必ずしもコンピュータ内部に記憶する初期形状にはふさわしくないと考えられるので、この部分に変形操作処理を入れることにより、ポーズを付けた形状でクレイモデルを作成し、コンピュータ内部に入力した後に、変形処理で初期形状にふさわしい形を作成する方法論を取ることが必要である．

5.3 質感の作成

作成された形状データに対して質感を作成する段階で、特にテクスチャーマッピングなどに代表されるマッピングテクニックが重要であると考えられる。質感作成上特にこの部分に関して統合化された作業工程の確立と道具の作成が必要となる。

6 処理詳細

それでは、実際の技術的トピックスに関する内容を紹介したい。

6.1 形状入力

形状をクレイモデルから直接入力する必要性は前述したが、実際の作業がどのように処理されるかを示したい。

既に作成されたクレイモデルを入力する方法論として我々がとっているアプローチは、３次元ディジタイザを用いた手法である。この手法のいままでの問題点は、クレイモデル上に最終的に入力するデータの為のメッシュを書き込む処理で

あった。実際クレイモデルにテーピングするなどしてメッシュを作成し、さらに頂点番号まで書き込んでから入力を行うなどの処理を行っていたが、これでは、短時間での入力は期待できず、かなりの忍耐と根性を必要としてた。

我々は、クレイモデル上をスタイラスペンでなぞることにより一定間隔以上離れたポイントを自動的に入力する手法を採用している。入力されたポイントをポリゴン化する処理に関しては、後処理とし、ある程度自動的に三角板の生成を可能としている。このアプローチは、三角板自動生成手法が機械的な処理であるので、万能ではないが、かなり作業を軽減することに成功している。

他のアプローチとしては、上述の手法で頂点データのみを高精度でコンピュータ内に入力し、既に作成した三角板の頂点座標の移動のあたりデータとして使用することも多い。この場合は、非常に粗い頂点データを入力し、三角板自動生成機能、もしくは、直接的に三角板を作成する機能で三角板ポリゴンデータを作成し、この形状データを細分化することにより、頂点座標移動前のデータを生成する。そうして作成されたデータを前述した超高密度頂点データを参照しながら変換していく。この処理は、ループ状に何回も実行され、必要なところで、パラメータを変更し、最終的に期待する形状に徐々に収束させていく。頂点データの入力後は、ほとんどの部分を自動化でき、オペレータは、コンピュータが形状を変化させていくのを観察し必要なところで調整を行うだけで良い。結果的に大幅な作業の軽減を実現できた。

これらの処理を複合的に利用することによって、キャラクタを一体入力するのに約2時間程度で完了させることが可能となった。これらの手法により、回数多く贅沢な試行錯誤が実現できるようになり。結果的に非常に説得力のある力強い形状をコンピュータ内部に構築するのに有効となっている。

6.2 変形問題

キャラクタのアニメーションに於て必要になる変形処理を、ある統一された考え方で処理することは、あらゆるレベルで重要なポイントとなる。

我々は、変形という処理を2つのレベルで取り扱うこととした。これは、最終的に必要になる表現の分析から得たものである。まずキャラクター全体を大局的に変形するレベルがある。このレベルでは、キャラクタの振付け時に必要になる種々の変形を処理することになる。第二のレベルは、上述で変形された形状表面をさらに微妙に変形させるレベルである。キャラクタが振付けによって変形した後に、例えばしわや、筋肉の盛り上がり等の変形を必要とした場合は、第二のレベルで処理する。

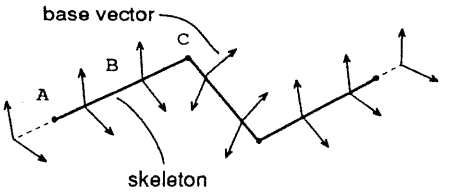
変形をこのような2つのレベルで取り扱い、キャラクターの振付け処理に沿った、アニメーションに有効な変形処理として、我々は独自にスケルトン座標系［2］という変形操作用の特殊な座標系をベースに考えている。

このアプローチは、主に第一のレベルの為に考案されたが、この特殊座標系の上に第二のレベルの変形を取り扱えるような仕組みを構築した。

スケルトン座標系は、キャラクターにアニメーションの段階で直感的に処理を考えられるようにするために、まずキャラクター内部にスケルトン（骨）を必要とする。骨は、グラフィクスユーザーインターフェイスを供えたエディタで作成され、キャラクタ表面の形状（ポリゴンデータの頂点）はこのスケルトンに対しある関係で接続（コネクト）される。結果的にスケルトンの移動によって、形状表面がつられて変形するという処理で変形処理を実現することが可能となった。

6.2.1 スケルトン座標系

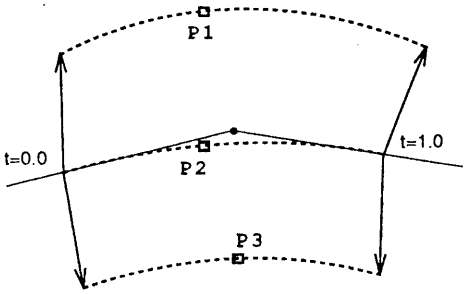
スケルトンは下図Aで示すデータで構成されている。ベースベクトルと呼ばれる2対のベクトルはスケルトンの近傍の空間を3つの領域に分割している。それぞれの空間をA,B,Cとする。この3つの空間の外側の部分に関しては、このスケルトンにコネクトすることは出来ない。通常は複数のスケルトンデータを連結してモデルを作成することになる。下図Aの場合は，3つのスケルトンが連結されている.



図Aスケルトンデータ

3次元空間中の1点p（x、y、z）はスケルトンデータによって区切られた3つの空間のうちのどこかにコネクトされるか、もしくはこのスケルトンには、コネクトされないかのどちらかの状態をとることになる。もしも、コネクトすることが可能な場合は、A,B,Cのどの領域にいるかを覚えておく。

ある領域について各ベースベクトルの先端を通る三次曲線をそれぞれ考える（図B）.



図Bベースベクトル間曲線

このときｔ＝０の時に片方のベースベクトル上の点を差し、ｔ＝１の時に反対側のベースベクトル上の点を差すような曲線を使用する。３本の曲線について上述のｔの値を０から１まで変化させると曲線上の点ｐ１、ｐ２、ｐ３も曲線上を移動することになる。このｐ１、ｐ２、ｐ３が作成する平面を考える。ある領域に属すことが解っている空間中の点は、ｔが０から１のあいだのある値の時に生成される平面上にのることが解る。この値をｓｚとする。ｔ＝ｓｚの時の曲線上の点の座標をｐ１’，ｐ２’，ｐ３’とすると、ｐ１’とｐ２’で作成されるベクトルＵとｐ３’とｐ２’で作成されるベクトルＶとの合成で空間中の点ｐを定義することが出来る。これを式で表すと

$$P=a*U+b*V$$

となり、このときのａをｓｘとし、ｂをｓｙとする。

これで、点ｐ（ｘ、ｙ、ｚ）はスケルトン座標値（ｓｘ、ｓｙ、ｓｚ）に変換された。つまり、形状表面がスケルトンにコネクトされたことになる。

実際の形状表面の３次元座標は、スケルトン座標値の３次元座標値への変換で求めることが可能である。

6.2.2 デルタコネクトデータ構造

スケルトン座標系を使用した変形計算では、スケルトンが動くような大きな変形指定に関しては、うまく取り扱うことが可能であるが、実際のキャラクターを考えてみると、骨は動かないが形状表面には微妙な変形が起こる場合がある。例えば筋肉の盛り上がりや、しわ等の変形である。これらの処理を上述のスケルトン座標値の使用による単純な計算だけで処理することはむずかしい。

これらの微妙変形操作を処理するためにスケルトン座標系内での座標値の差分データを考える。このデータのことを、我々は、デルタコネクトデータと呼んでいる。このデータだけを名前を付けてディスク上にセーブすることが可能だが、このことは、微妙な変形そのものに名前を付けて、名前での変形の参照を可能とするものである。すでに作成された差分データどうしの演算も考えることが可能であるので、この差分データをうまく作成することができれば、形状表面の微妙な変形操作が、スケルトンの空間内の位置に依存しない形で実現でき、変形を名前で参照し、割合で指定することが可能となった。

差分データの作成に関しては、形状入力の項で示したいろいろなアプローチで直接的に生成することが多い。現在のところ説得力のある差分データの作成には、欠かせない手法となっている。

逆にスケルトンの空間中の位置関係を基に、上述の差分データを自動生成することも考えられる。このことは、差分データの自動生成機能として構築することが可能である。

6.2.3 トポロジーの違う形状間のメタモル

スケルトン座標系を介した変形処理及びスケルトン座標系上での差分データを取り扱うデルタコネクトデータなどを使用することにより、形状の変形に対してかなり有効な手段を構築することができた。しかしながら、これらの考え方のベースになっているのは、おおもとの形状を変形させてアニメーションを作成するという考え方である。このことは、キャラクターアニメーションにおいては、あまり無理のない前提であるかと思われたが、実際には、いろいろと問題が発生する場合がある。

おおもとの形状を変形させてアニメーションを

作成していくということは、アニメーションの途中の段階で生成されるオブジェクトのトポロジー（頂点数や、ポリゴンリスト）は全ておおもとのオブジェクトにひとしくなる。作品の演出によっては、アニメーション過程においてトポロジーが変化する場合が予想される。分離したり、新たに融合したりするおおきなレベルでのトポロジーの変化については、最初から作成方法を検討したほうが良い場合が多いと考えられ、その場合には、メタボールなどのアプローチが有効な場合が多い。しかしながら、作業工程上メタモルしてく物体の形状はほとんど無理のないレベルで完成されているが、データ上トポロジーが違うという局面は、驚くほど多く出現してしまう。

この問題を解決するためには、ほとんどおなじ形状を表現しているが、トポロジーが違うという物体に対し、強制的にどちらかのトポロジーを移植する手法をとる。このことにより制作作業工程で発生するデータ精度の変化によるトポロジーの違いを機械的作業で１つのものにあわせていくことが可能となった。

具体的なアプローチとしては、３次元空間内での頂点や、ポリゴンの位置関係をみて、基準のデータのポリゴンリストを保存し、その頂点座標値をあわせる側のオブジェクトから移植する手法をとる。非常にシンプルな処理である。なお、同等の処理を形状入力時にも多用することは、すでに記した。

6.2.4 シミュレーションによる変形

アニメータが変形して欲しいと考えて、変形処理をコントロールするアプローチについて我々のアプローチを紹介してきたが、我々が、使用しているもう一つのまったく違ったアプローチがシミュレーション計算による形状のコントロールである。

現在のところおもに全ての形状データを一旦生成し終った後に、後掛けでシミュレーション変形をかけるアプローチをとることが多い。おもに、形状表面がブヨブヨしている状態などのコントロールは、おもにこの手法を用いている。

例えば既に生成された形状データの動きより，その状態での力を求め，その力を制御することにより，ブヨブヨした状態をシミュレートしている．

6.3 質感作成

キャラクターアニメーションにおいて，形状の作成の次の段階で重要になるのが，作成した形状のに対する質感の作成である．我々が取っているアプローチは，ポリゴンデータで形状を作成する手法なので，今日使用されている質感の作成に関する色々なアプローチを取り入れることが楽であるというメリットがある．

いろいろなアプローチの中で，基本とも言えるテクニックに形状表面へのマッピングのテクニックがある．これには，画像データを貼付けるテクスチャーマッピングや，法線を貼付けるバンプマッピング，質感パラメータを貼付けるアトリビュートマッピングなど，多種のテクニックが既に考案されている．これらのテクニックをキャラクターアニメーションで有効に活用するためには，データの貼付け方にある工夫が必要である．我々は，物体表面に貼付けるデータと，物体表面とのあいだにある対応関係データのことを，コーディネイトデータと呼んでいる．このコーディネイトデータの考え方に工夫が必要というわけである．

有機的複雑な形状をしたキャラクター表面へのマッピングを考える場合，コーディネイトデータに関しては，もはや，平行投影的な考えや，極座標的考えはとることができない．なぜならば，このようなマッピングにおいてもっとも重要な要素は，アニメーション中にキャラクター表面のマッピングがずれないことであり，形状がどんなに説得力のある完成度で作成されていても，マッピングずれが発生してしまうと途端にリアリティーは減少してしまうからである．

このマップずれの現象を回避するには，ポリゴンの頂点ごとにマッピングデータ上の２Ｄ座標を持たせる必要がある．コーディネイトデータをどのように考えるかということは，言い替えれば，物体形状の頂点データに与えるコーディネイトデータをいかにして作成するかということである．

それでは，我々がどのようなアプローチでコーディネイトデータを作成し，マッピングデータを生成しているかについて示したい．

6.3.1 コーディネイトデータ生成

有機的で複雑な形状をもった物体に対し，きれいにマッピングデータ用のコーディネイトデータを作成することは，自動化が難しい極めて人的な判断を要する処理である．なぜならば，作成される形状はどのようなものであるかまったく予測がたたず，さらに，その形状が最終的なアニメーションの中で，どのような演技をするかは，１００％演出に委ねられてしまっているからである．したがって，カメラがキャラクターに非常に接近するために，マップデータを部分的に高解像度にしておきたいなどという要求を臨機応変に満たせなければならない．

最終的に貼付けるマップデータは，いわゆる通常の画像データの形式に沿った２Ｄデータであることが望ましい．もしも２Ｄデータであるならば，スキャナにより取り込んだスキャン素材や，２Ｄペイントシステムで作成リタッチした画像データを素材として使用することが可能であり，このことは，作品のクオリティーアップと，作業工程の短縮化のいみで，極めて重要である．したがって，我々は，マッピング素材を２Ｄのもっとも普及しているとおもわれるデータ形式に沿ったものにした．

このマップ素材を形状に貼付ける関係を考える場合に，我々が取っているアプローチは，開きオブジェクトを作成しそれを板に貼付けてコーディネイトデータを作成するという手法である．前述のエディタは，この部分を強化する目的で設計された．この考え方は，動物の毛皮などのサンプルをマッピング素材として活用する上でも威力を発揮すると思われる．

既に作成した形状から開きオブジェクトを作成するには，簡単に示すと２つの工程を処理する．１つは，メスをいれて裂け目を作成する処理．もう一つは，そうして作成された形状を任意の平面まで開く処理である．この２つの工程は，ループ上に繰り返され，作業の進行状況に応じて必要になった場面でメスをいれ，革をなめすような感覚で開きオブジェクトを作成していく．

そうして作成された板上のオブジェクトをある平面に対して貼付けると，その平面上の２Ｄ座標がコーディネイトデータとなる．

前述の演出上のいろいろな要求は，すべてこのコーディネイトデータを作成する段階で解決することになる．最終的に形状表面にデータが貼ついた場合のマッピングデータの引き延ばされ方に関しては，コーディネイトデータの板の上での密度でおおよそ把握できるので，必要に応じて，ポリゴンの頂点データのポジションを変更することによって期待するデータに近付けていく．この開きオブジェクトの作成に関する試行錯誤には，種々の形状編集コマンドが非常に有効となる．

6.3.2 ３次元ペインタ

このようにして作成されたコーディネイトデータをもとにマップデータの作成に取りかかる．コーディネイトデータを作成する段階で既にスキャンされた画像をあたりとして使用している場合などは，その画像をもとにしてリタッチ作業でマップデータのクオリティーを上げることが一般的である．このような場合には，スキャン素材を２Ｄペイントシステムに読み込みコーディネイトデータを頭に描きながらの修正処理でもかなりのクオリティーのマッピングデータを作成することができる．現在のほとんどのマッピング素材のリタッチは，このようなアプローチで作成されていると考えられる．

しかしながら，上述のようなアプローチが簡単に取れないような場合も考えられる．このような場合に有効な手段として，３Ｄペイントシステムと呼んでいるエディタを開発した．

これは，プラモデル等を実際に作る場合の色付けの作業をコンピュータ内部でシミュレートするもので，３次元表示された物体に直接ペイントすることができ，自由に視点データを変更して，塗りたい面を見たい角度から見ながら，好きな筆で色をつけることができるものである．

このシステムにより，有機的物体の表面に大胆に色を付け，この段階での試行錯誤が直接的な操作で繰り返せることとなった．

このシステムの構築に当っては，グラフィックエンジンの使用を前提として，一般的に普及しているエントリーモデル程度の比較的低速マシンへのインプリメントに成功している．

7 実験作品

7.1 変形モデルの検証 In Search of New Axis

有機的形状の変形にスケルトン座標系モデルが

有効であるかどうかの検証の為に制作した．実験して検証する必要のあった項目としては，パイプ曲げ的変形と，分岐している形状の変形についてであった．この項目に関しては，良い結果を得られた．

物体の形状は，簡単な形状を細分化して作成した．各々の物体は，三角板にして約５万から７万ポリゴン程度のデータを持つ．

表面の質感に関しては，仮想球を使用したリフレクションマッピングを用い，シェーディングモデルには一般的なブリンシェーディングを使用した．レンダリングに関しては，スキャンラインアルゴリズムのレンダラーを使用した．レンダリングに要した時間は，sun4上で一枚約１時間程度であった．最終的には１５台前後のUNIXマシンで分散させて計算して約１週間程度かかった．

１９８９年の４月の中旬に制作をおこなった．作品の秒数は１分１０秒．

7.2 表面変形モデルの検証
In Search of Muscular Axis

この作品では，変形形状の表面で，微妙なしわや，筋肉の盛り上がり等を，スケルトン座標系の上で作成コントロールする手法（デルタコネクト）の検証を行うことを技術的目的としている．

説得力のある形状の作成と，さらにその形状の上でのリアルな盛り上がり等の変形．そして，微妙な変形に関するデルタコネクトの処理においては，変形自体に名前をつけ，その名前での演算処理などにおいて，良い結果を得られた．

形状の入力に関しては，クレイモデルからの直接的入力を行い，物体は１．５万から５万ポリゴン程度の容量をもっている．質感の作成に関しては，仮想球によるリフレクションマッピング及びブリンシェーディングを用いた．また変形の一部において，後掛けのシミュレーションを行っている．

レンダリングは，スキャンラインアルゴリズムを使用したレンダラーを使用し，レンダリングに要した時間は，iris4D-20上で約２０分程度であった．

１９９０年の春に制作をおこなった．

7.3 マップのコントロールに関しての検証
In Search of Performing Axis

この作品では，有機的な形状へのマッピングに関する色々なレベルでの新しい試みを行った．

実際には，パーティクルシステムによるスプレーの映像と，スキャンラインレンダラの生成するキャラクタの映像をＺコンポジットし，さらに，パーティクルシステムと，３Ｄペインターによるマッピングデータの作成手法の確立と検証を行っている．変形の一部に後掛けのシミュレーションを行っている．

レンダリングは，スパークステーション２上で１枚やく５分から１０分程度の時間を要している．

制作は１９９１年の春から夏にかけて行われた．

8 今後の課題

今後の課題として，現在以下の２点について作業中である．

1）スケルトンデータのアニメーションに関して
2）シミュレーション系の総合的なコントロール及びエディタの開発

9 参考論文

［1］島本，「多面体の中点変位による近似曲面生成手法」，第５回NICOGRAPH論文コンテスト，P120-P129

［2］加藤，「スケルトン座標系」，第５回NICOGRAPH論文コンテスト，P138-P147