卒 業 研 究 報 告

題　　目

3 次元コンピュータグラフィクス

指 導 教 員

原 央

報 告 者

井上 雄介

平成　　13 年　　2 月　　9 日

高知工科大学 電子・光システム工学科

目次

1. Introduction

1－1.CG の歴史

コンピュータグラフィックスの始まりは、サザーランドのスケッチパッドにあると言われている。これは 1963 年に開発されたもので、ブラウン管(CRT)すなわちディスプレイを利用し、対話形式でグラフィクス図形を取り扱った最初のシステムであった。

一方コンピュータを用いた映像作成の試みは､1960 年代の初頭にアメリカのベル研において始められた。その目的は、エンターテイメントではなく、まさにエンジニアリング分野における必要性により開始されたもので、姿勢制御装置を装備した人工衛星の角度変化の運動を、迅速かつ正確に計算して、ディスプレイ上に映像表示する、宇宙開発のためのシミュレーションだった。

1960 年代には、初歩的なものながら、簡単な CG システムが作成されだした。この頃のシステムはワイヤーフレーム表現をとっており、形状の判別が難しかったため、多くの隠線消去の手法が研究された。また、ワイヤーフレーム表示から、サーフェス表示へと変化するにつれ、スムーズシェーディングの研究も進み、1970 年代初頭の 1971 年にグローシェーディングが、1973 年にフォンシェーディングが考案された。

1970 年代からは、TV 放映や、CG 制作会社の設立がはじまった影響で、CG が一般の人々にも注目されだした。この頃から、当初の目的だった、軍事や生産技術の向上のための技術から、徐々にエンターテイメント分野への活用も始まったと言える。1970 年代の後半になると、テレビのみならず、映画における特撮場面にも CG が利用されるようになってきた。とくに 1977 年につくられた作品『スターウォーズ』においては、戦闘場面の特殊撮影を CG により作成し、従来の特撮技術では不可能であった迫力ある場面が展開された。また 1979 年には、NASA が木星に探査衛星を接近させた際の衛星フライトシミュレーションが CG 技法で作成され、テレビ放映された。

さらに 1982 年に上映された『トロン』は、興行的には失敗したものの、CG ブームを巻き起こす要因となった作品である。先駆性のある人々の関心を集め、現在でも残る CG の有力制作会社がこの頃に多く作られた。

1970 年代に CG はアメリカにおいて全盛期を迎えたが、80 年代からは日本においても CG ブームが起こってきた｡たとえば､本格的な CG プロダクションである JCGL（コンピュータグラフィックラボ）やリンクス（東洋 現像所の資本による CG プロダクション）が誕生し、まさに CG の実用化時代を迎えたと言える。その後多数 の CG プロダクションが発足し、多くの作品がつくられた。それにより、従来実写が主体であったテレビコマーシャルにも､CG により作成された映像が続々登場し､また、番組のタイトルやニュースのバックタイトルなどにも、CG の技法を用いて作成された映像が数多くみられるようになってきた。またこの時代になると、CG やタイトルのみならず、事故状況の再現（たとえば航空機事故のフライト再現）にも有効に利用

され始めた。

CG が実用化時代に入った最大の原因は、超 LSI の出現、および画像処理技術のハードとソフトの両面における急速な進歩である。1970 年代には、CG 映像は大型コンピュータを駆使するか、または 1 本の作品に数ケ月をかけて、多大な労力を費やしてつくられていた。現在では，高性能なワークステーション(特にグラフィクス・ワークステーション)が安価に普及しつつあり、またパソコンのグラフィクス機能の向上にともない、誰もが手軽に CG 作品を作成できる時代になった。

2. 3 次元コンピュータグラフィクス

2－1. 3 次元コンピュータグラフィックスの概略　(図 1-(a),(b),(c)参照)

3 次元コンピュータグラフィクスは、計算機内に定義された形状モデルをディスプレイに表示することである。この処理過程を効率よく行う方法が、これまでに開発されている。

図 1-(a)は、ワイヤーフレーム表示といわれ、多面体の総ての稜線を表示したものである。図 1-(b)は、本来見えない線を消去して表示したものである。この表示法によりややわかり易くなる。さらに図 1-(c)は、見えない面を除去して、可視面に濃淡を付けて表示したものである。

3 次元コンピュータグラフィクスを作る時、まず、形状を計算機に記憶しなければならない。これは、頂点の 3 次元座標と、そのつながり（稜線および面）を記憶する。次に 3 次元物体を 2 次元平面上に表示する必要がある。そのため投影が行なわれる。次に、図 1-(b)のように見えない線を除去する。この処理を隠線消去という図 1-(c)のように見えない面を除去することを、隠面消去という。また、光源を与えて、面上の各点での輝度を求める処理をシェーディングという。

2－2. 画像の生成過程　(図２参照)

モデル（形状および面の性質）が定義された後、視点（またはカメラ位置)、視野範囲、光源の位置および種類が与えられると、スクリーン上の各画素で色を表示する。

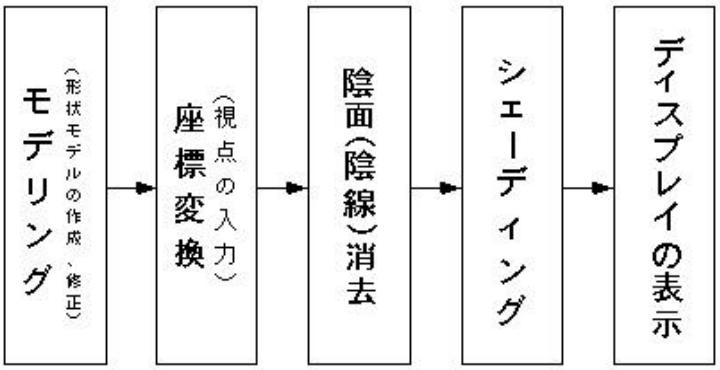


図 2

図 2 に 3 次元 CG の生成手順の概略を示す。データを入力し、その形状を計算機内に内部表現する過程をモデリング(Modeling)とよぶ。座標変換は、移動、回転および投影を含む。線画として表示する場合には隠線消去が施される。一方、

陰影画像なら、隠面消去がされ、最後にシェーディングされて、ディスプレイに表示される。座標変換、隠面消去、シェーディングを含めてレンダリング(Rendering)と呼ばれる。システムによっては、形状データを作成するプログラムであるモデラーと、表示するプログラムである、レンダラーとに分割してあるものも多い。

2−3. モデリング

2−3−1. 3 次元形状の表現技術

表示したい形状を模型化する過程をモデリングと言う。コンピュータ上での幾何学的な 3 次元形状の表現には、その記述方法の違いから、ワイヤーフレームモデル、サーフェスモデル、ソリッドモデルの 3 種類に大きく分類できる。

(1)ワイヤーフレームモデル (図３参照)

3 次元形状を数値化するためのもっとも基本的な方法は、形状を 3 次元座標系の中での座標値として表すことである。形状を線分で表し、それぞれの線分の両端の頂点を X,Y,Z の 3 次元座標系で表現することで数値化を行う。このように形状を数値化してしまえば、これらの座標値に対してさまざまな計算を行うことにより、形状の移動や変形など、座標変換による自由な形状の操作ができるようになる。このように線分の集まりで表された形状をワイヤーフレームモデルという。

(2)サーフェスモデル　(図４参照)

ワイヤーフレームモデルが3次元形状を頂点と頂点を結ぶ直線で表現したものであるのに対し、サーフェスモデルは 3 次元形状をその表面を構成する面の状態で表したものである。形状の表面をポリゴンといわれる細かい多角形平面に分割し、その平面を構成する頂点の座標値によって形状を表現する。

しかし、表面を細かな平面で覆うという方法では滑らかな曲面の表現には限界があるため、格子状に分割した平面をもとに自由な曲面を表現する方法が使われる。これは曲面パッチと呼ばれ、その代表的なものに自由曲線をもとにしたベジェ曲面や B-スプライン曲面などがある。

このように、形状の表面をより滑らかにしかも正確に表現する方法が開発されているが、サーフェスモデルという表現方法では物体の表面形状の状態しか表していないため、物体を中身の詰まった状態として扱うことが難しい。また、面のどちら側が物体の内部であるかの情報をもっ

ていないから、必ずしも閉じた多面体を構成している保証がない

(3)ソリッドモデル　(図５参照)

ソリッドモデルとは、物体を中身の詰まったものとして表現する方法で、サーフェスモデルに面の方向の情報を加え、閉じた立体としたものである。

ソリッドモデルの代表的な表現形式に CSG(Constructive Solid Geometry)がある。CSG は、3 次元形状をいくつかの物体の組み合わせによって表すもので、この組み合わせを集合の足し算や引き算などの演算によって行う。(図 28 参照)

組み合わせの基本となる 3 次元物体は一般にプリミティブ(primitive)と呼ばれ、球、円柱、円錐、立方体などがある。CSG は、数式で表されたこれらの基本形状に基づいて複雑な形状を表現していく。

この手法の長所は、形状を論理式だけで表現できるためデータが比較的小さくてすむということである。また、プリミティブが数式で表されているため、拡大・縮小などの変形も容易である。短所としては、形状に自由度があまりないことがあげられる。組み合わせをいくら複雑にしても、物体のそれぞれの面はやはりプリミティブそのものであるため、自由曲面を含むような複雑な形状の表現は困難である。

2－3－2. モデリングに使われる材料

(1) ポリゴン

① ポリゴン　(図６参照)

最も基本的かつ、本質的な CG におけるモデリング用の手段である。直接ポリゴンを編集できないソフトウェアでも最終的にはポリゴンに変換してレンダリングされることが多いため、大半のソフトウェアがポリゴンを使って CG データを制作しているといえる。

ポリゴンとは 3 次元空間上の位置を示す点の座標と、それらを結ぶ順番によって多角形を表現したものである。曲面を表現するためにはポリゴンとポリゴンとの接続情報も付加される。

点と面の法線(どちらを面が向いているかをあらわすベクトル)情報が付加されていることもある。通常は面の法線は自動的に計算で求められてシェーディング時に使用されるが、時に編集する必要が出てくる。法線の編集による結果はシェーディングしてみないと確認できない。また各点ごとに色情報を持ったデータ構造を持つこと

ができるものもある。バーテックス・カラーと呼ばれるこの頂点毎の色はシェーディング計算や、3D ペイントツールによって決められる。バーテックス・カラーは一般にシェーディング計算やマッピング計算をレンダリング時に行うよりも速いため、高速な描画を目的とした処理系で有用である。ポリゴンモデリングの利点はどんなトポロジーを持ったものも自由自在に作ることができるということ、同じ形状を表現するのに最適なポリゴン数にまで極小化することができること、モデリングしたものがそのままレンダリングに渡されるまでの処理時間も最小ですむことなどがある。

ポリゴンによるモデリングは一点一点、ポリゴンの一枚一枚を最終的にはコントロールする必要があるため、ソフトウェアの提供している機能によって使い勝手が大きく左右される。ポリゴン編集機能に優れたソフトウェアでは数十万ポリゴンのオブジェクトまで思い通りに扱うことができるが、貧弱なソフトウェアでは数千ポリゴンがせいぜいである。

② サブディビジョン・サーフェス　(図７参照)

ポリゴンの一つ一つの点を編集しなくてはいけない煩雑さを無くし、しかもどんなトポロジーのものでも作ることができるのがサブディビジョン・サーフェスである。サブディビジョン・サーフェスは分割のレベルを指定することで非常に滑らかなポリゴンの発生から、軽いデータまでを同時にもつことができる。いろいろな形状や大きさのポリゴンオブジェクトに有効であるため、フレキシビリティが高い。必要に応じて必要なところに頂点を増やしたり、消したりすることができる。ポリゴンの長所と NURBS パッチの長所を合わせたかのような特徴を持ち、現在最も進んだモデリング手法である。サブディビジョン・サーフェスのコントロールポイントをアニメーションすることでかなり融通性の高い変形アニメーションも作れる。

(2) パッチ

① パラメトリック曲面

ポリゴンによる直接的なモデリングではどうしても角張った形になりやすい。滑らかな形を作るには多くのポリゴンに分割しなくてはならず、そうなると変形や修正が非常にしにくく、扱いが難し

い。この問題をクリアするために自由曲線関数を使ったパッチ曲面と呼ばれる一連の技術の発達を見てきた。基本的なパッチは二組のスプラインの間を直線的な補完をすることによって一枚の四角形上のポリゴンメッシュを自動生成し、曲面の形状をコントロールする。中割の細かさはポリゴンを発生させる時に決めることができるため、パッチでは縦横方向のステップと呼ばれるパラメータが使える。スプライン関数の選び方によって様々なタイプがあり、それぞれに特徴がある。

② B-スプライン　(図８参照)

最も一般的に知られたスプラインで、曲線の生成自体もあばれが少なく、扱いやすい。制御点が実際に生成される点よりもかなり離れたところに置かれるため、実際に点やポリゴンを生成してみないとどんな曲面になっているのかわかりにくい、慣れるまでモデリングしにくいという特徴がある。一方局所的に角が立った所を作ろうとすると、制御点を増やさなければいけないので、必要以上にポリゴンが増えてしまう。

③ ベジェ　(図９参照)

ベジェ曲面の特徴は生成される曲線が制御点そのものを通り各制御点に接線ベクトルを指定できるため、非常に細かいレベルまで曲線の特徴をコントロールでき、また局所的に角を立てることも容易にできる。一方この接線ベクトルを使った制御は多くの制御点がある場合には作業が煩雑になりやすく、B-スプラインに比べるとデータも重くなりやすい。編集のしやすさはソフトウェアの設計による部分が多い。

④ カーディナル　(図 10 参照)

カーディナルスプラインは B-スプラインの滑らかさと、曲線が制御点上を通るという分かりやすさを兼ね備えたスプラインである。感覚的には操作しやすいが曲線が若干あばれやすい。

⑤ NURBS　(図 11 参照)

スプライン曲線の関数によってそれぞれくせがあり、どれも一長一短だったのだが、NURBS の導入によってかなり使いやすさがあがった。NURBS は B-スプラインの拡張で、少ない制御点で滑ら

かな曲線を生成でき、しかも各頂点におけるウェイトをコントロールすることで局所的な曲面の固さも制御できる。パッチどうしのスティッチやマージ、トリムなどの曲線どうしの演算処理が可能で、それらの機能を使ってモデリングの幅を広げることができる。ウェイトコントロールができるが、意識して編集しなければ通常の B-スプラインのように簡単に編集することもでき、必要に応じてウェイトの編集をすることで見た目の扱いやすさが高い。一方ウェイトをきちんとコントロールしようとするとモデリングソフトのインターフェースによって使い勝手が大きく左右されることになる。数値入力からウェイトをペイントする方法まで様々な方法がソフトウェア毎に用意されている。

(3) メタボール　(図 12 参照)

メタボールとは、空間内に存在する「互いに影響し会うガス状の球」と考えることができるもので、これを用いれば非常に複雑な 3 次元形状を表現することが可能である。メタボールは、濃度の中心と、その周りに実体として存在する球、さらにその外部にあるそのメタボールの有効範囲とで構成される。メタボールは、中心を基準とした連続した濃度の値の減少関数をもっており、2 つのメタボールの有効範囲が重なると、それぞれの関数を足し合わせた新しい関数ができる。よって、1 個のときには球と変わらないが、2 個以上になると互いに融合し合って複雑な形状になる。このようにして、たくさんのメタボールを組み合わせることで、複雑な形状を表現することができる。

メタボールの長所は、滑らかな融合を表現できるという点である。これによって人体や顔のような有機的でしかも複雑な形の表現が可能となった。一方で、現段階では球や楕円球を基準としたものしかないということが短所としてあげられる。複数の楕円体を構成して任意の形状をつくっていくため、どうしても楕円体が融合したような形状になりがちであり、平面や稜線、角をもった幾何学的な形状の表現には限界がある。

(4) その他

直接モデリングの手段ではないが、モデリングに関わる周辺技術。

① ボリュームレンダリング　(図 13 参照)

空間そのものを視覚化する手法である。レンダリング時に光の進行経路上の各空間における様々な属性から色を求める。属性は透過率や色情報であったり、様々である。ポリゴンやパッチなどのよう

な目に見える外形そのものではなく、空間の属性の集合としてものの形を表現することができる。そのため、半透明な物体の内部や、人間の想像の及ばない複雑なトポロジーを持った物体なども表現することができる。一番よく使われるのが煙の表現や、塵や埃にあたって見える光の道筋の再現である。直接空間における各点の属性を編集することは難しいため、大抵の場合にはプロシージャルな方法で値が求められる。滑らかな形状や表現を再現するにはサンプリングの密度を細かくするほか、値の補完に技術を要する、物体の表面と光の衝突だけで計算を行う通常のレンダリング計算に比べて、計算を行わなくてはいけない点が増大するため、計算コストが高い。

② ディスプレイスメントマッピング　(図 14 参照)

　マッピングの一種で、物体の表面に凹凸情報をマッピングすることで形状を変化させることができるものである。テクスチャによって制御できるため、場合によっては一つ一つの点を編集するよりもずっと感覚的に、ペイントのような気分でモデリングすることができる。ただしテクスチャの解像度に合わせてオブジェクト自体も細分化しないと、必要なディテール以上にデータ量が発生してしまう危険性もある。また効果の程をレンダリングしてみないと確認できない歯がゆさもある。テクスチャという間接的な方法を使ってモデリングすることができるため表現の幅が広がる。テクスチャをアニメーションすることによる変形表現や、ディスプレイスメント・マップ以外のテクスチャとの連動もやりやすい。テクスチャをレイヤー化することでモデルの修正もレイヤー化することができる。プロシージャルなテクスチャを貼ることで、プロシージャルなモデリングをすることもできるなど、可能性の大きい方法である。ディテールを逐一造形していくよりも、精度の高いディスプレイスメント・マップ用のテクスチャを描くほうがモデリングもしやすく、修正も楽な場合がある。

③ ノイズ、フラクタル　(図 15－(a),(b)参照)

　ノイズ関数やフラクタル関数は、プロシージャルな方法で生成される値を用いて様々な目的に利用される。ノイズ関数はランダムであるが連続的な値の変化を持つ関数一般のことを指す呼び名で、フラクタル関数はその中でも再帰的な周波数成分を持つもののことを指す。ノイズ関数やフラクタル関数を利用する方法としては一番

一般的なのはソリッドテクスチャである。３次元空間中の位置を与えることで値を計算し、その値をテクスチャの色情報に置き換えてレンダリングに利用するものである。またこの関数をモデリングに利用することもできる。一つはポリゴンで作られたオブジェクトにノイズ関数やフラクタル関数によって計算される値で再分割を行い、表面の凸凹を自動的に生成させる方法。これによって岩山や山肌のような自然なゴツゴツした感じをモデルに与えることができる。もう一つの利用方法はメタボールなどのプリミティブの配置やボリュームレンダリングにおける値の算定に使うもので、人間が手では作れない複雑なオブジェクトをプロシージャルに生成することができる。

④ パーティクル　(図 16 参照)

パーティクルシステムは大きさをもたない粒子の動きを制御するもので、炎や煙などを粒子の集団として表現する。またこれらの無機的な物質だけでなく、魚や虫など動物の群れの動きのシミュレーションにもこのパーティクルシステムが応用されている。

2－3－3. モデリングにおける基本的制作手法

本来、ものを作る方法は、ものの形を見た時に人間が認識する方法や、ものの形を発想する方法に近いものが理想である。例えばコップを人間が見た時に漫然と全体の形を何となく認識するのではなく、断面が丸い感じの円筒形をしているとか、厚みの変化は上から下に向けて広がっているとかということを感じとりながら、形状の特徴を認識する。それならばコップをモデリングする時にこのような認識の仕方、ものの形の捉え方をそのまま手順に置き換えることができれば一番直感的に作ることができる。また自分で立体物をデザインする場合にも、細かいところから大まかなところまで一気に写真を撮るかのように発想されるのではなく、大まかな構造やシルエット、また特徴的なラインなどから発想が進み、やがて細部に至るように進むことが多い。そういう順序で発想したのであればそのような手順でものを作れるのが一番理想的である。しかし現状の CG ソフトウェアを用いたモデリング手法は、道具の特徴を主体として考えられてきた。バーテックスやポリゴン、パッチといった各オブジェクトを使って、いかに感覚的に作れるかという模索の中でいく通りもの方法が用いられている。

(1) ロフティング(ポリゴン、パッチ)

オブジェクトの形状をいくつかの断面で捉えて、そのつなぎ合わせによって立体の形状を作る方法がロフティングである。この方法はものの形を断面の形状、つまり曲線で捉え、それが変化していく経過をいくつかの曲線で表し、最後にそれらの曲線間に面を貼ることで立体化させる。

ロフティングは主にパッチを使って作る場合に多用されるが、ソフトウェアがその機能を提供する場合にはポリゴンメッシュでも使うことができる方法である。

どんなに断面の形状が変わっても一本の曲線で表現できる場合には一つの連続した曲面として扱うことができるが、途中から断面の形状が複数の曲線に分かれたり、またくっついたりするようなトポロジーを持った物体では、その変化があるところでの接合が問題になる。ポリゴンモデルでは点と点の対応さえ決めればつなぐ方法はあるが、パッチでは一度オブジェクトをそこで切り分けて、後でつなぐ算段をしなくてはいけない。

この方法でモデリングする場合にはものの形を断面の曲線で捉えることが重要になる。イメージする形状や今から作ろうとするものを観察して、そこにどんな断面が見え、それがどのように並んで面を構成しているのかを見極めながらモデリングを行う。

① リボルブ／スウィープ (図 17 参照)

リボルビングはある断面を表現する曲線を回転させながら面を発生させてつなぐことで立体化する方法である。回転体と呼ばれる、軸を中心として対照的な形をした物体のモデリングに適している。

② エクストルード (図 18 参照)

エクストルードは押し出しとも呼ばれ、断面を表す曲線をある軸上に複数並べてつなぐことで形状を得る方法である。細長く断面が一定の物体を作るのに使われる。

③ スキニング (図 19 参照)

上記の二つは断面の外形をあらわす曲線は一つで、それを回転させたり、押し出したりしながらたくさん並べて、形を作るものであるが、スキニングは断面を表す曲線自体をたくさんユーザー

が定義しておき、それらの何本もの曲線の間をつなぐことで形状を求めるやり方である。断面形状がどんどん変化するが基本的には断面の連続によって表されるようなトポロジーを持ったものであれば自由度の高いモデリング手法である。

(2) プリミティブ形状の変形(ポリゴン、パッチ)　(図 20 参照)

ポリゴンでもパッチでもよく使われるスタイルのモデリング手法が、あるプリミティブ形状のオブジェクトを変形させていくことで望む形を得るものである。ソフトウェアが提供する形状の物体をあらかじめ作成し、その頂点やコントロールポイントを編集することで形を変えていき、最終的に目的の形になるようにする方法である。この方法で用いられるプリミティブは球、円筒、グリッド(四角辺)であることが多い。最初から面が存在し、その変化を目で見て確かめながら形をつめていくことができるため作りやすい。この方法では編集途中で点を増やしたり、減らしたりすることもできる。そのため、最初は少な目の点のプリミティブからスタートし、必要に応じて点を増やしてやるのがよい。ポリゴンではトポロジーにとらわれず、局所的に点を増やしたり、減らしたりできるが、そうするとポリゴンの流れが最初のプリミティブと変わってくるため、ポリゴンの流れが乱れないように注意しなくてはならない。パッチでは行や列毎にしか点を増やしたり、減らしたりできないため、最初のプリミティブで決まっているトポロジーのまま変化していくことになる。パッチでは異なるトポロジーに展開するにはトリムやブレンドといった別の形状とつなぎ合わせる手法を使わなければならない。

この方法でモデリングする場合にはプリミティブの変形でどこまで形を出せるのか、どこから別の方法を使うのかを見極めながら使う必要がある。

(3) 引っ張り出し(ポリゴンのみ)　(図 21 参照)

ある単純な形状をもとにし、その必要な面を引っ張り出しながら、あるいは穴をくりながら、面を増やしていき、必要なディテールを加えて行く方法である。これも基にする形があることでは上記の方法と同じであるが、最初は極めてシンプルな形からスタートし、どんどんトポロジーを変化させていくことで求める形に近づけていく点が違う。この方法では中心部から周辺に向かって様々な枝分かれをしていくような立体形状を作るのに便利な方法である。しかしトポロジーの

変化を伴うため、パッチでは使えない。感覚的には粘土を引っ張り出しながら作り上げていくのに近く、ポリゴンであるため、後でいくらでも必要な箇所に点を増やすことができる。

この方法では当然ながら最初の形から最終的な形へと変化させながら作り上げていくプロセスが大事なので、作るものを観察する時や発想する時にそのような作業の流れをイメージする必要がある。

(4) マージ、くりぬき(パッチのみ)　(図 22 参照)

ポリゴンの場合、上で述べた方法どうしの組み合わせが自由にでき、いくつでも違う方法に乗り換えていくことができる。しかしパッチを使った場合、作り方によっては途中で別の方法を使ったものに切り替える必要が出てくる。無理してある形をひねくり回していても、いたずらに点を増やし、重いデータになっていく割に面が綺麗につながらず、しわがよるなど良くない影響もでてくる。パッチオブジェクトでは基本的に一枚のパッチが 1 オブジェクトであるため、複数のパッチをつないで一つの物体を構成する方法を用いなくては複雑なオブジェクトは作れない。こうした場合にあらかじめ自分が作りたい形がどのようなパッチの分割によって構成されるのか計画立てておくことが望ましい。後からパッチの枚数を増やしていくのは、ポリゴンを増やしていくのに比べて、データ量の増加が激しく、偏った比重になることもあるからである。これらの複数のパッチをつないで形を作るための機能としてマージやトリムなどがある。マージは複数のパッチを隣接する境界で一つのパッチにつなぎ直すもので、トリムはあるパッチ上に別の曲線を投影させ、そこから接合面や穴を作るためのものである。その他部分的にパッチの機能を共有化する機能などもある。複雑なトポロジーを持ったオブジェクト作るためには、これらパッチどうしの編集機能は不可欠である。NURBS パッチでは計算上の融通性の高さからこうした機能が豊富に使える。

(5) 変形

直にモデルの形を編集するのではなく、変形機能を使うことで形状を変化させ、望む形に近づけていくこともある。これらの手法の多くは、モデルを包む 3 次元座標を歪ませる方法で、その歪ませ方によって様々な機能に分けられている。

① ツイスト

ある軸を設定し、その軸を中心にして断面を回転させることに

よってモデルを「ねじる」変形を施す。軸は必ずしも直線とは限らず曲線を軸に変形することもできる。

② ベベル　(図 23 参照)

変形ではないがモデリングでよく使われる機能の一つにベベリングがある。これは角張った形としてベースを作っておき、後からその角を丸めるための機能である。機械の部品や工業製品などに良く使われる。

③ テーパ　(図 24 参照)

ある軸に沿って、位置によって軸に垂直な方向の縮小、拡大を行うことができる機能。

④ ラティス　(図 25 参照)

モデルを 3 次元格子で囲み、その格子を変形させることで空間を歪ませてモデルを変形させる機能。FFD とも呼ばれる。中に含まれているモデルの複雑さに関係なく、空間を歪ませるため、格子を構成する点の細かさによってどのくらい細かい変形を行うかを決めることができる。一方 3 次元格子上の点の編集は立体的に点が配置されているため、あまり細かい格子点を設定してしまうと変形するうちに訳がわからなくなる危険性がある。

ソフトウェアによってはこの問題を解消するため、3 次元格子点に限定しない、別の形で変形を制御できる方法が提供されているものもある。

⑤ サーフェスフィット　(図 26 参照)

もとになるモデルのある平面を別に用意した曲面上に投射することで変形する機能。ある曲面に沿って別の形状が貼り付いているように見える物体を作る場合や、複雑な形状を大まかな曲面に合わせて変形させるような場合に用いる。この場合にも変形に使う曲面をアニメーションさせることでモデルの変形を制御することができ、幅広い表現に生かすことができる。

⑥ スプラインフィット

もとになる物体の軸を別に作った曲線に沿わせることで変形する機能。細長い形をしていて全体として曲線をなしているよう

な物体を作る時に用いられる。また変形に使う曲線を編集することで容易にアニメーションすることができ、管状の物体の造形には有用である。また曲線上を移動させることもできる。

⑦ ベンド　(図 27 参照)

変形の仕方自体はスプラインフィットに近いが、これもある軸上の位置によって、回転する角度を変化させることにより、オブジェクトを曲げる機能。

(6) ブーリアン演算　(図 28－(a),(b),(c),(d)参照)

形状と形状を重なり具合によって別の形を求める機能がブーリアン演算である。ブーリアン演算では原理的には空間として閉じた内部を持ったソリッドモデルどうしでしか計算できない。ただし例えば空間が開いていてもポリゴンには裏と表が定義されているのでそれを利用して擬似的な演算を施すことができる場合もある。

この演算によって求められた形状は別のオブジェクトの特性を合わせたものになるため、通常ポリゴン化される。もとになったオブジェクトのポリゴンの接続の仕方などが異なるため、法線の連続性が保てず、シェーディングがきれいにならないといった問題点が発生しやすい。これを避けるためにはもとのオブジェクトの法線をそのまま引き継いで保持しておくことが考えられる。レンダリング時にブーリアン演算をすることができるソフトウェアではこの問題を解決することができるが、レンダリングしてみないと結果が見えないという欠点もある。

① 掛け算

二つのモデルの互いに交わった部分を新たなオブジェクトとして作成する演算。

② 足し算

二つのモデルの双方を包括する曲面を作成する。

③ 引き算

二つのモデルの片方から反対側のモデルで削り取った形を返す。どちらからどちらを引くかによって形が異なる。削られたところは削ったモデルを裏返しにした断面を持つ。

2－4. レンダリング表現技法

2－4－1. 隠面（線）消去アルゴリズムの分類

隠線消去や隠面消去について、多くの手法が開発されており、その分類も種々考えられる。判定を行う空間で分類すると、画像空間アルゴリズム、物体空間アルゴリズム、および優先順位アルゴリズムに分類できる。画像空間アルゴリズムおよび物体空間アルゴリズムは、それぞれ判定をスクリーン上を基本に行うか、3 次元空間で行うかである。優先順位法は、3 次元空間で前後関係を求めておき、スクリーン上で遠方のものから順に重ね描きする方法である。

① 画素単位

スクリーン上の各画素について可視となる物体を抽出する方法である。この分類に属す方法として、Z-バッファ法とレイトレーシング法がある。前者は、表示しようとする面の視点からの奥行き（Z 値）を各画素ごとに蓄える方法である。記憶されている Z 値と表示しようとする面の Z 値を比較することで可視面を判定する方法である。後者は、各画素と視点とを結ぶ視線を考え、視線と交差する面のうち最も手前の面を表示する方法である。

② 走査線単位

この手法は、各走査線と視点を含む平面によってスライスされた２次元的なウィンドウを考え、このウィンドウと交差する物体の面との交線を抽出し、視点から見える部分を表示するものである。面の一様性を利用できるので、アルゴリズムはやや複雑であるが、記憶容量は少なくてすむ。アルゴリズムの性質上、ラスタ型 CRT 上への表示に適しており、プロッタのようなベクトル方式のものには不向きである。なお、前述の Z-バッファ法や次の領域細分法を スキャンライン単位に行うものも開発されている。

③ 領域単位

表示画面を順次細分する方法であり、各々のウィンドウについて、システムがあらかじめ用意したアルゴリズムを用いて物体の面または線分が判別できるまで、この細分を繰り返す。その度合は、システムがもつアルゴリズムの複雑さに依存し、簡単なものほど細分割の必要がある。

④ 画面単位

あらかじめ表示物体の総てについて、相互位置関係を調べ、視点からの奥行き情報をテーブルに準備しておく方法である。視点から遠い順に面を塗りつぶす操作を繰り返すことにより、遠方のものは手前のものに塗りつぶされ、最終的には手前の面（即ち可視面）がスクリーンに表示される。 この方法は油絵を描くように下から順に重ね描きすることからペインターズアルゴリズムとも呼ばれる。また、スキャンライン単位に重ね描きすることもある。この方法には、物体データ作成時に適当に分割された 3 次元空間領域に各々の物体を割り付ける方法と、視点が与えられた後に奥行きの順位を求める方法がある。

2−4−2. 隠面処理

3 次元の物体を眺めたとき、裏側にあったり物体の陰になっていたりして見えない部分がある。このような、見えないはずの面を描く対象から外し、可視面だけを描くようにする処理を隠面処理と呼ぶ。

(1) Z ソート法(奥行きソート法)

Z ソート法はラスタスキャン型ディスプレイの塗りつぶし表示の特徴を活かし、視点から遠い順に面を重ねながら次々と描いていくことによって見えるべき面が最終的に残るというもので、面ごとの奥行き値 Z に面の重心を用いることが多く G ソート法とも呼ばれる。隠面処理の中ではもっとも簡潔な部類のアルゴリズムである。しかし、遠い面の判断が正確にできさえすれば簡潔ではあるが、互いに交差する(貫通する)面どうしでは、面を細分する前処理を行はなければならない。また、面の大きさや位置関係によっては、重心値のみでは正確な前後判定ができないことがある。

(2) Z バッファ法

Z バッファ法は、表示しようとする面の Z 値（視点からの奥行き）を各画素に蓄える方法である。各々の面を透視投影面に投影し、その面が占める画素の位置に、その面の Z 値を格納する。この際、すでに格納されている面があれば、その Z 値と比較し、小さい方の面の情報を記憶する。

この方法はアルゴリズムが簡単で、ハードウェア化に向いている。また、大きな記憶容量を必要とするが、メモリの低廉価に伴いグラフィクワークステーションでリアルタイム処理可能なハードとして標準装備

されることが多くなった。各画素に Z 値を求めるので複数の面が相互に干渉する場合でも適用できる。しかし、影や反射屈折の処理はできない。また、エリアシングの問題（面の境界が階段状のギザギザを生じる）が残る。

(3) スキャンライン法

視点とスクリーン座標上の横1列の画素(スキャンライン)を通して物体を結ぶ平面を、スキャンライン平面という。スキャンライン平面は、画面の縦方向の画素数と同じ枚数が存在する。視点は無限遠に設定するので、スキャンライン平面はすべて 3 次元直交座標の xz 平面に対して平行なものとして扱われる。

スキャンライン法による隠面処理は、各スキャンライン平面に含まれるポリゴンを調べ、視点から画素を結んで伸ばした視線上で、最も近いポリゴンをその画素に描いていく方法である。最も近いポリゴンを構成している物体が透明か半透明ならば、次 に近いポリゴンについても調べて画素の色成分を決める。

スキャンライン法では比較的短時間で計算可能だが、対象物が複雑でポリゴン数が多い場合には、多量のメモリが必要となる。Z バッファ法とは異なり、必要メモリ容量は画像の解像度に左右されない。前述のように光を透過する物体についても表現ができる他、アンチエイリアシングも可能。光線が物体に遮られた場合にできる影の表現も可能である。

(4) レイトレーシング法

視点から出た視線は、物体にあたるまでのび続け、そして、物体に当たった視線は、跳ね返る『反射光』と、物体の中を通る『屈折光』にわかれる。この２つに別れた光線をそれぞれに追跡し、さらに物体にぶつかった場合は反射光・屈折光に分かれる。これを繰り返すことにより、１つの視線は複数の光線に分岐する。最終的に視点に飛び込んでくる全ての情報を収集するのがレイトレーシング法である。

レイトレーシング法は、光線を分岐させるだけの比較的簡単な手法ではあるが、反射・屈折・陰影の表現等、極めて高品位の画像を生成できる。しかし、光線が何度も分岐することで計算が膨大な量になり、時間がかかることが最大の弱点と言える。この弱点を軽減するために、分岐する回数を抑制したり、物体との交点を効率よく計算する方法が研究されている。

2－4－3．シェーディング

(1) 光源

現実に物体にあたっている光は、光源からの直接的な光(直接光)、壁や物体どうしで反射した光(間接光)、空気で拡散した光(天空光)などが合わさったものである。CG では直接光が照明モデルとして扱いやすく、光の要素としては色と強度が取り上げられる。色は RGB の光の三原色で指定することが多く、それぞれの色の要素ごとに光の強度を指定する。

直接光の種類には、点光源、平行光源、スポットライト、線光源、面光源、多面体光源、天空光などがある。

① 点光源　(図 33 参照)

位置と色と強度を持ち、すべての方向に均等に照射される。理論的には、光源からの距離の 2 乗に反比例して減衰する。平行光源に比べ光源と物体との距離を感じさせることができ、絵のスケール感をコントロールするのに役立つ。影の大きさも同時に変わり、平らな面に対しては緩やかな明暗のグラデーションを作る。

② 平行光源　(図 34 参照)

方向と色と強度を持つ。基本ライトの中では最も回り込みがあり、太陽光のような無限遠からの光をシミュレートするために使う。位置を持たないことから、一般に距離による減衰はないものとする。したがって、距離を感じさせないフラットな照明に向いている。影の大きさは小さい。

③ スポットライト　(図 35 参照)

強い指向性を持った点光源として扱われ、位置と方向と照射角度(指向性の強さ)と色と強度を持つ。また、スポットライトが当たった部分は光軸の位置が最も明るく、辺縁にいくほど暗くなる。指向性が強いほどこの減衰は急激で、逆に指向性が弱くなるほど一般の点光源の特性に近くなる。

(2) 反射係数

物体の表面色が同じでも材質によって異なって見えるのは、各物体それぞれが光に対して持っているいくつかの性質(反射係数)が異なるためである。反射係数は、アンビエント(環境光)、デフューズ(拡散反射光)、スペキュラー(鏡面反射光)と呼ばれるデータで、シェーディングに大き

く関わる物体の材質感を決める要素である。

① アンビエント(環境光)

周囲の物体からの反射光など、弱い光の成分である。部屋の中で、ライトが直接当たらない壁などでも、ある一定の明るさになっているのは、周囲の物体に反射された光のせいである。宇宙空間のように周囲に物体が存在しないと、このような反射は起こらず、三日月のような現象が見られる(太陽に向いていない月面は真っ暗になり見えない)。

CG でアンビエントを計算するのは難しく、普通は環境光という光源を設定して、明るさを補うことが行われる。

② デフューズ(拡散反射光)

光源から入ってくる光が、全ての方向に均一に反射するのが散反射光である。つやのない紙、ゴム、壁などはこのような反射をする。一般に物体の色というのは、白色光を当てたときに拡散反射した光の色のことである。

③ スペキュラー(鏡面反射光)

入射した角度のちょうど反対側（反射角）に強く反射する光である。拡散反射光は全方向に均一に反射するが、鏡面反射光では一方向に鋭く反射する。強く反射した光の先に視点があるとき、 視点から見てその場所は回りの場所と比べて極端に光ることになり、いわゆるハイライトになる。また、ハイライトの色を決定付ける要素でもある。

(3) シェーディング

隠面処理後の物体の各面に色を付ける処理をシェーディングという。単に固有色で面を塗りつぶすだけでなく、光の影響を計算して影の部分を表現するので、「影付け(シェーディング)」と呼ばれる。

① ランバートシェーディング

すべての入射光は完全拡散反射を行うと仮定した計算で、つまりデフューズだけでスペキュラーの計算を行わない。その分計算が速く、全くスペキュラーを使わない場合は非常に高速。しかし、最近ではあまり使われていない。

② フォンシェーディング

反射光の角度に応じて、ある分布で拡散反射することを仮定して求めたスペキュラー成分を持つ。現実の表面をモデル化していないので、ハイライトを極端に広げたりすると実在感が失われて見えることがある。

③ ブリンシェーディング

大抵の 3DCG ソフトで使用できる中で最も精密なスペキュラー計算をしている。表面の法線のばらつきを考慮に入れた計算式を用いるため、ある種の質感においては十分説得力のあるハイライトが得られる。

④ グローシェーディング

スムーズシェーディングの 1 種。曲面近似ポリゴンの各頂点の色を比例配分して、ポリゴン内の色を求める。一般にフォンシェーディングよりも計算速度は速いが、ハイライトがきれいに付かないなどの欠点がある。

⑤ スムーズシェーディング

グローシェーディングのように頂点の色を比例配分するのではなく、頂点の法線ベクトルを比例配分することによって、ポリゴン内部の色を求める。

2－4－4. マッピング

自然界は無数の物体で構成されており、個々の物体はそれぞれ個有の模様(texture)による質感をもっている。このような複雑な情景を写実的に描こうとすると、無限大に近い記憶容量と計算時間が必要である。これを避けるための主な手法として種々のマッピング手法が開発されている。

(1)テクスチャマッピング　(図 39 参照)

2 次元画像を 3 次元形状の表面に壁紙のように張り付けることによって、物体表面の模様などを表現する。2 次元画像の情報は、物体の色として使う他、法線、反射・屈折方向の色、透過率などシェーディングの際のあらゆるデータとして使われる。

現在のコンピュータグラフィックスではサンプリングされたデータ

を扱うために写像の方向が問題になる。まず、テクスチャ画像上でのサンプリングの間隔が生成画像の画素の大きさと一致しないことが問題になる。テクスチャ画像のサンプリングの間隔が、生成する画像上で画素の大きさより大きいときに、マッピングされない画素が生じてしまう。逆に、テクスチャ画像のサンプリングの間隔が小さい場合には、その場合の処理が難しく、かつ計算時間も長くなる。

(2)バンプマッピング　(図 40 参照)

凹凸を見かけ上で表現するマッピング手法である。テクスチャマッピングがスカラマッピングであるのに対して、バンプマッピングは、(u,v)で表される平面の柄の濃淡を法線の傾き要素として陰影計算を行うベクトルマッピングである。したがって、光源が正面にあって陰の出にくい部分や光の当たらない部分では効果がない。また、座標そのものが変化するのではないため、極端な凹凸表現は物体の縁では不自然になる。

(3)リフレクションマッピング　(図 41 参照)

レイトレーシング法を使わずに鏡面の映り込みの質感を表現する手法である。映り込む画像は、あらかじめその物体から見た周囲の景色を調べ、入力あるいは生成させておく。そして、物体面から反射した方向にある景色をマッピングすれば、あたかも鏡のように見える。擬似的な方法であるから多少の矛盾は発生するが、レイトレーシングの処理コストを考えれば、効果的な手法である。

(4) リフラクションマッピング

レイトレーシング法を使わずに透明物体における屈折された景色を表現する手法である。リフレクションマッピングと同様な手法で屈折光に対して適用することにより、比較的リアルな透明物体を簡易的に表示することができる。写り込む周囲の景色の方向が正反射方向から屈折方向に代わるだけで、リフラクションマッピングとほとんど同様の手法である。しかしながら、周囲の景色が屈折によって目に到達するには、少なくとも 2 回の屈折をしているために、少々注意深く見れば矛盾が発生する。

(5) ソリッドテクスチャマッピング　(図 43 参照)

木材の木目模様などをテクスチャマッピングで作ろうとすると、隣り

合った面のつなぎ目で柄をあわせることが難しい。また、そのために各面ごとのテクスチャを用意することも非効率的である。そこで、3 次元的なテクスチャを計算によって発生させ、これを物体に対して空間的にマッピングする方法が用いられる。しかし数学的な関数によって作り出すため、どんな模様でもできるというわけにはいかず、木目や大理石など限られたものしかない。

3. アニメーション(ムービー)制作

3－1. 概要

(1) 目的

3 次元コンピュータグラフィックスのアニメーションを作ること。

(2) 制作場所

株式会社 高知デジタルスタジオ

(3) 制作期間

2 ヶ月

(4) 使用した機器、アプリケーション

- Illustrator
- Photoshop
- After Effects
- 3D Studio MAX

3－2. 使用したアプリケーション紹介

(1) Illustrator

印刷用、Web 用ベクター・グラフィックスを作成するためのソフトウェア。画像を点と点を結ぶ関数として扱うドロー系のソフトウェアで、変形を繰り返しても画質が悪くならない。その時に応じて拡大・縮小されるロゴやチャートなどの図の作成に適している。

◆作品で利用した主な機能、目的

3DCG 制作においてはテクスチャの作成に用いることが多いと思われるが、今回は、最後の「Thank you」の文字をこのツールを用いて作成したくらいで、これでテクスチャを作ることはなかった。

(2) Photoshop

写真のレタッチ、画像編集、カラーペイントを行うソフトウェア。ピクセル(画素)という小さなタイルで作られたデータを扱うペイント系のソフトウェアで、写真を扱うのに適している。1 ピクセルごとの編集が可能だが、変形を繰り返すと画質が悪くなる。

◆作品で利用した主な機能、目的

テクスチャを作成するために使用した。今回使用したテクスチャの内、一部のテクスチャは自分で Photoshop を用いて作成したが、ほとんどは 3D Studio MAX（以下 MAX）に標準で入っているテクスチャやインターネット上に置いてあるフリーテクスチャを使った。図 44 と図 45 は MAX に標準で入っているテクスチャの例である。2 枚とも砂漠のシーンで使用したテクスチャで、図 44 は砂漠の砂、図 45 は建物の外側の壁の模様である。

(3) After Effects

単純なモーションから複雑なモーションまで、様々なモーションを作成、制御、および追跡することができる。モーションを分離して、ビジュアルエフェクトを作成したり、画像のぶれをなくしてモーションを滑らかにしたり、スクリプトを実行してアニメーションを自動化することができる。

◆作品で利用した主な機能、目的

主に、MAX で作成したシーンをレンダリングする時に TIFF というフォーマットの連番の画像ファイルで出力し、それをつなげるために使用した。

また、MAX の同じシーンで別々にレンダリングしたオブジェクトを後から合成するために使用した。しかし、同じシーンなら別々にレンダリングせず、すべてのオブジェクトを一度にレンダリングしても最終的な映像は同じであるし、後から合成する手間もない。なぜこのような方法を取るのかというと、これは例えば、背景は動かず、その中のキャラクターだけが動き回るようなシーンが 100 フレームあった場合、シーン内のオブジェクトを全部一度にレンダリングすると、動かない背景を 100 枚もレンダリングすることになる。そのためこのようなシーンでは、背景は 1 枚だけ、その中で動くキャラクターだけを 100 フレーム分レンダリングして、After Effects で合成するという方法を取る。また、例えこのシーンで背景が動いたとしても、別々にレンダリングして後で合成する。これは、レンダリング後、背景だけ、もしくはキャラクターだけ修正したい場合があるからだ。修正後は、背景だけ、もしくはキャラクターだけをまたレンダリングすればよい。このようにシーン内のオブジェクトを別々にレンダリングして、後から合成することによって、全体の制作時間を大幅に短縮することができる。図 46～48 参照。図 48 のシーンを作るとき、背景は動かないので、図 46 を１枚で十分だが、バイクと人は動くので､図 47 のように分離してシーンの長さ分のレンダリングが必要。

他には、MAX でうまく影付けできなかったもの（影を地面に映すことができなかったもの）の影を作成したり(図 49)、文字を入れたりするために使用した。文字自体は Photoshop や Illustrator で作成して、それがフェードイン、フェードア

ウトしていく様子や最後の「Thank you」のように筆で書かれていく様子、光が文字の中を走る様子を After Effects を用いて表現した(図 50)。また、作品中、主人公がバイクで建物の外に飛び出すシーンがあるが、そのシーンで背景にある太陽は、After Effects の Lens Flar Pro というプラグインを利用して表現した(図 51)。

(4)3D Studio MAX

3DCG 作成用のソフトウェアであり、以下のような機能がある。

① ポリゴン、パッチ、スプラインパッチ、NURBS と多くのアプローチが可能なモデリング機能。
② モディファイヤによる、オブジェクト指向の履歴編集機能。高度で様々なコントロールが可能なパーティクルシステム。
③ アニメーションボタンとトラックビューによる直感的で柔軟なアニメーション機能。
④ 数多くのシェーダーと豊富なアンチエイリアスアルゴリズムを搭載し、自由度が高く細かな調節が可能なマテリアル。
⑤ レンズ効果、被写界深度といった様々なエフェクトのリアルタイムプレビューが可能な高速・高品質レンダラー。
⑥ ポストエフェクトの追加や簡単なムービー編集も行うことができるビデオポスト。
⑦ 内部にまでアクセスが可能で、マクロレコーディング機能も搭載した MAX スクリプト。
⑧ 全ての機能をユーザーの好みに合わせて自由に使用することができるカスタマイズ可能なユーザーインターフェース。

これら全ての機能を完全に 1 つのアプリケーション上で扱うことができるため、MAX を使えば 3DCG 制作を行うほとんどの作業を単体で完結させることができると思われる。

さらに、より高度な機能が必要な場合にはプラグインを利用して、様々な機能を自由に追加、高機能化させて使用することができる。MAX はプロ・アマ含めてユーザー数が多いので、フリーのプラグイン数も他の 3DCG ソフトウェアと比べて非常に多く、優秀なものも数多い。特に最近では SimCloth や MAX Havok Lite、FireStorm など、有料であっても決しておかしくないものが多数フリーウェアとしてリリースされている。また、MAX 自体の開発に関わっているメーカーやプログラマーが独自にそういったプラグインをリリースしている場合もあるので、しっかりした安定性を備えているものも多い。

◆作品で利用した主な機能、目的

1) プラグイン

プラグインとは、アプリケーションに標準で搭載されていない機能だが、使いたい時に後から付け足すことができるソフトウェアのことである。

① FireStorm

極めてリアルな炎、煙を作り出すプラグイン。ボリュームとしてギズモだけでなくパーティクルを用いる機能、マッピング機能、ライトによって照明されたり、影を落としたり、影を受けたりする機能、背面にあるライトを遮る機能などが搭載されている。しかし、β版なので一部不安定なところもある。

図 52 の飛んでくるバイクを包んでいる炎とそこから伸びている煙と図 66 の爆煙はこのプラグインで作った。

② Textporter

複雑な形状のオブジェクトのテクスチャを描くための下敷きを作成してくれるプラグイン。単にポリゴンを展開するだけでなく、選択したオブジェクトのマテリアル ID を選択して展開したり、法線によってカラーをつけたりもできる。

作品中では人の顔のテクスチャを作るために用いたが、MAX で作った顔のオブジェクトにこのプラグインを割り当てると図 53 ができる。Photoshop でこの図を目安にして描いたものが図 54。図 55 は図 54 を人の顔に貼ったものである。このようにして顔のテクスチャを作ると図 55 のように眉や唇の位置がずれない。

2) モデリング

MAX のモデリングの仕方には、ポリゴン、パッチ、NURBS の 3 種類があり、それぞれにメリット、デメリットがあるので用途によって使い分けることが重要になる。

① ポリゴン

レンダリング速度、画面表示速度は 3 つの中で最速。

作品中の建造物やバイクは曲面が少ないのでこのモデリング方法で作成した。（図 56、図 57）

② パッチ

レンダリング速度、画面表示速度は、ポリゴンよりは遅いが、NURBS よりはずっと速い。

作品中の人物はこのモデリング方法で作成した。図 59 では、左側の面(パッチ)が貼ってあるオブジェクトの形は常に右側のスプラインだけのオブジェクトの形を参照するように設定してあるので、左側のオブジェクトで形を確認しながら、右側のオブジェクトのスプラインを変形して最終的な形状を作成する。（図 58、図 59）

③ NURBS

レンダリング速度、画面表示速度ともに非常に遅い。（図 60、図 61）

3) マテリアル

マテリアルエディタ（図 64）で様々なシェーダー、マテリアル、マップが用意されており、それらのパラメータをコントロールすることで様々な質感をオブジェクトに与えることができる。

4) パーティクルシステム

図 65 はプラグイン FireStorm を使って爆煙を作成しているところである。赤い玉がパーティクルであり、これらの動きが煙の動きになる。レンダリングすると図 66 のようになる。

5) ボーン

ボーンは人間のように関節のあるオブジェクト用に骨格を作る機能で、動きのあるオブジェクトをより簡単にアニメーションするために使用される機能。（図 67、図 68）

① スキンモディファイヤ

ボーンを腕や足、胴体などに割り当てて、それらのサーフェースをボーンの動きに合わせて変形させる機能。

② インバース・キネマティクス（IK）

エンド・エフェクタのターゲット位置を指定すると、インバース・キネマティクス(逆運動)アルゴリズムが各ジョイントの回転角度を計算しエンド・エフェクタをターゲット位置に移動させる。インバース・キネマティクスは複雑な階層オブジェクトの操作に理想的な機能である。標準的には手足または図形の各部をそれぞれ適切な位置に配置し、これら

をグループ化して階層を構築する。

6) モーファーモディファイヤ

2 つ以上のオブジェクト間で実行することによりシェイプを変えるアニメートされたオブジェクトを作成できる。単一のオブジェクトが形を変えつつあると見えても実際にはモーフ処理がローカルの座標系に相対的に頂点の位置を一方のオブジェクトにおける配列から他方のオブジェクトにおける配列に変換している。したがって、モーフオブジェクトの構成に選択された全てのオブジェクトは同じ数の頂点を持っていなければならない。

この機能は主に人の顔の表情の変化に使われる。たくさんの表情を持たせたいなら、たくさんの表情が違う顔のオブジェクトを用意しなければならないが、それぞれのオブジェクトを形成している頂点の数は同じでなければならないので、まず、基になる顔を 1 つ作成し、他の表情の顔はそれをコピーしてそのコピーオブジェクトを形成する頂点の位置を移動して別の表情にしていく。(図 69)

3－3. 制作過程

① ストーリー、登場するキャラクター・建物のデザインをする。
(本当は、この段階で絵コンテを完成させる。)

② 登場するキャラクター、及び建物を MAX でモデリングして、質感を与える。
また、アニメーションさせる必要があるキャラクターにはボーン(骨)を入れる。

③ ②で作成したモデルを 1 つのシーンに集めて、ライト、カメラを追加して、アニメーションを付ける。
また、煙や炎などをパーティクルシステムとプラグイン使って作る。

④ シーンが完成したら、レンダリングする(スキャンラインレンダリング)。このときの出力形式は TIFF というフォーマットの連番の静止画像で出力する。

⑤ ④で出力した連番画像を After Effects でつなげる。また、1 つのシーンの中でいくつかのオブジェクトに分けてレンダリングしていた場合は、ここで合成する。

⑥　すべてのシーンにおいて④と⑤の作業が終わったら、After Effects で合成したすべての静止画像をレンダリングする。出力形式は AVI という動画像のファイルフォーマット。

⑦　最後に⑥出力した AVI ファイルをビデオテープに出力する。

3－4. 作品について

４月に高知デジタルスタジオ（以下 KDS）で研修を開始して、３ヶ月程 3D Studio MAX を使い、扱いにも慣れてきたところで、とりあえず何か１つ CG ムービーを作ってみようと思い作った作品である。

作品のストーリーは、砂漠の中にある要塞から主人公がアメリカンのバイクに乗って、宙に浮かぶバイクで追いかけてくる敵から逃げる、といったものである。爆発シーンや実際には無い乗り物などが出てくる SF アクション映画のようなものを作りたかったので、このような作品を作った。

制作中で一番苦労した部分は図 66 の爆煙のシーンだった。前述のようにこの爆煙は、パーティクルと炎のプラグインで表現しており、煙がトンネルから飛び出してきて壁で跳ね返るようにしている。パーティクルの球の動きで煙の動き方をシミュレーションしてから、レンダリングする。シーンの長さは約 2 秒であるが、レンダリング時間は 10 時間以上かかった。

また、できれば効果音などを入れたかったが、時間的な都合でできなかった。

3－5. オープニングキャラクターについて

最初に出てくる小柄で青い服のキャラクターは、高知県信用保証協会のホームページの「何なの君」というイメージキャラクターである（図 70）。KDS で 6 月～7 月頃、高知信用保証協会のホームページ制作がされており、私も、2D でデザインされた「何なの君」を 3D 化してアニメーションをつけるというかたちで手伝わせて頂いた。この作業で、キャラクターへのボーンの通し方、アニメーションの付け方を勉強することができた。

4．まとめ

4－1．制作結果

2 分程度の 3DCG アニメーションを作ることができた。

4－2．身に付けた技術

①3D Studio MAX をつかった 3DCG のモデリング及びアニメーション
②After Effects を使った映像の編集と合成、特殊効果
③Photoshop を使った画像の加工

4－3．感想

当初はこの作品の他に、まだもう１つか２つ、ムービーもしくは静止画を作るつもりだったが、思ったよりも時間がかかってしまい、１つしか作ることができなかった。また、この作品についても当初思い描いていたストーリーの半分くらいとなってしまった。その理由は、時間的な都合と私の技術的な都合もあるが、作業計画がかなり雑だったことが一番の原因だと考えている。全体の大まかなストーリーを頭の中に置いて、絵コンテもそこそこにしか描かず、ほとんど行き当たりばったりで作っていたので、今思えば大変能率の悪い作り方である。

５．謝辞

KDS で半年間、大変貴重な経験を積みました。ハイスペックなマシンと CG ソフトを使うことができ、しかも、その時々の私のレベルに合った仕事を KDS の方々が与えてくださり、そのおかげで、ずいぶんレベルアップすることができました。また、私も制作に協力した CM やホームページが、テレビで流れたり、インターネット上にあったりすると、ものを作ることの楽しさを改めて実感することができました。そして、念願の CG ムービーもたった 2 分程度ですが、作り上げました。これは、大きな喜びです。ただ、私自身もこの作品の完成度に満足できないのは、心残りです。この気持ちを忘れずにこれからもどんどん作品を作り続けていきます。この作品のリメイクをいつかするつもりですが、次回は、また別なものを作ってみようと思います。

ここでの経験は今後の私の人生に大きな影響を与えると思います。

KDS の久保さん、相澤さん、今村さん、野島さん、金山さん、橋田さん、岸上さん、東村さん、長瀬さんには大変お世話になりました。特に相澤さんには、CG 関連の本やビデオなどいろいろなものを見せていただき、また、CG 業界に関する話やデザイナーとしての生き方など興味深い話をよくしていただきました。そして、橋田さんには、CG 技術からプライベートに至るまで様々な面で面倒を見ていただき、最

後の作品制作の時もいろいろ助けていただきました。また、RKC プロダクションの皆さん、土佐ナビの皆さんにも大変お世話になりました。本当に感謝します。
ありがとうございました。

参考文献

デザイン編 CG 標準テキストブック　発行　財団法人画像情報教育振興協会
対策 CG 検定　コンピュータイメージ研究所　著

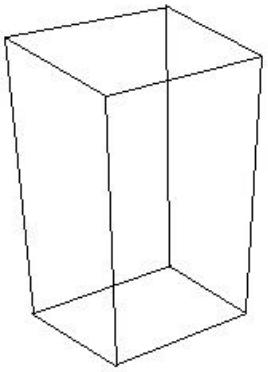


図 1－(a)

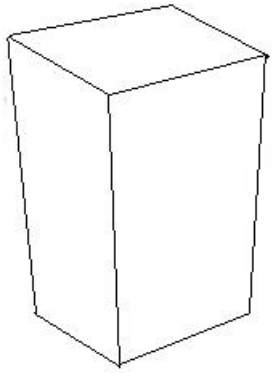


図 1－(b)

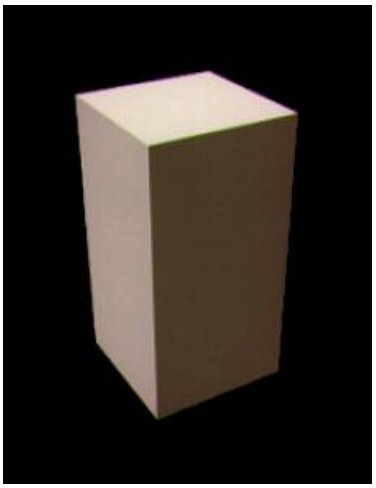


図 1－(c)

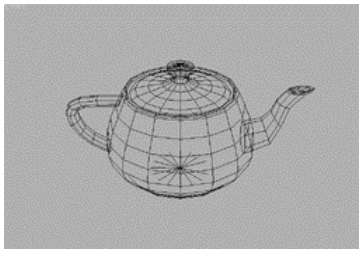


図 3

図 4

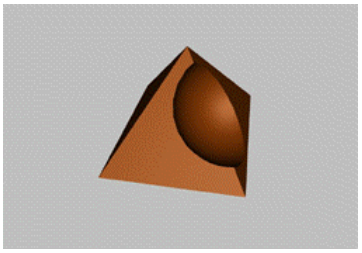


図 5

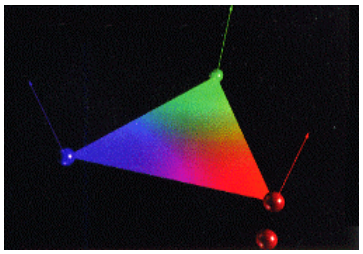


図 6

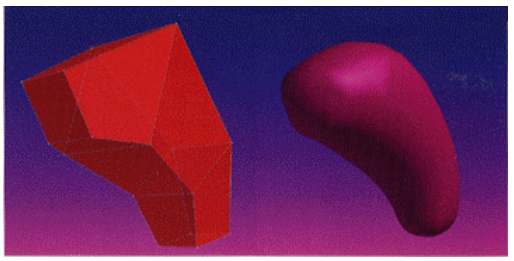


図 7　左がサブディブジョンサーフェス細分化前
右がサブディブジョンサーフェス細分化後

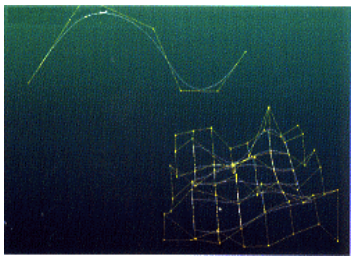


図８　B－スプライン

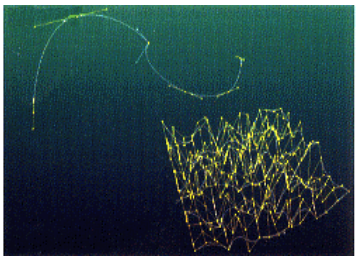


図 9　ベジェ

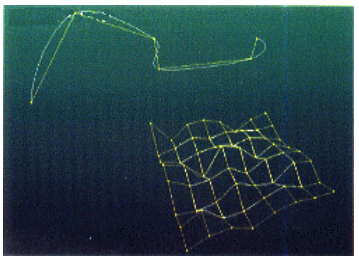


図 10　カーディナル

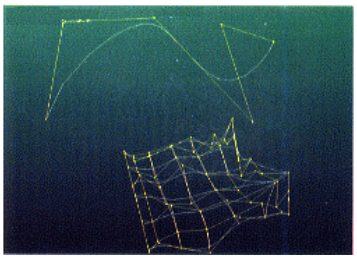


図 11　NURBS

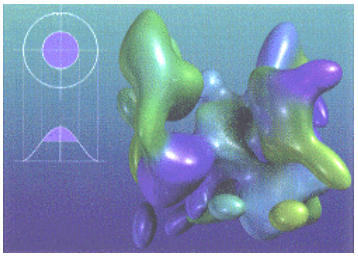


図 12　メタボール

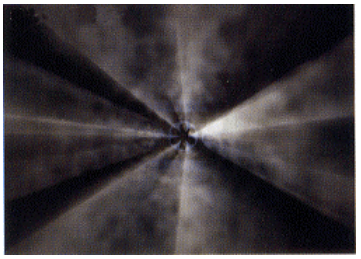


図 13 ボリュームレンダリング

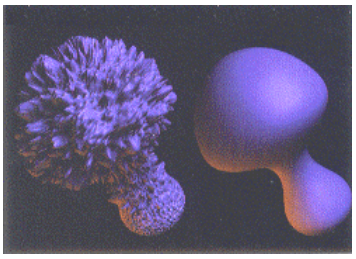


図 14
ディスプレイスメントマッピング

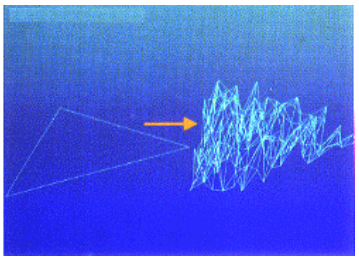


図 15－(a)
ノイズサブディビジョン

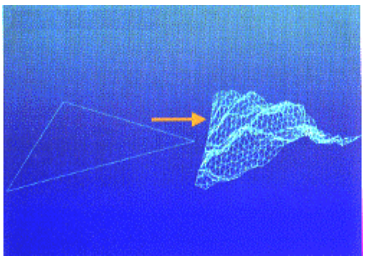


図 15－(b)
フラクタルサブディビジョン

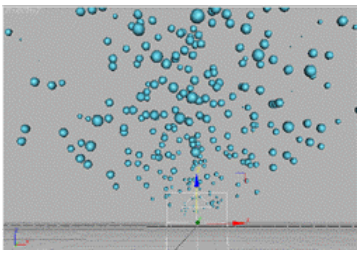


図 16　パーティクル

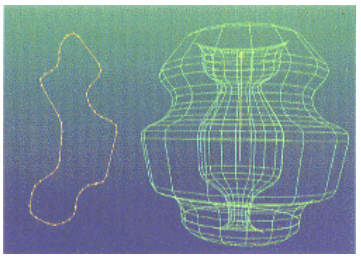


図 17 リボルブ/スウィープ

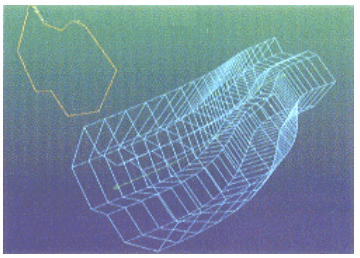


図 18　エクストルード

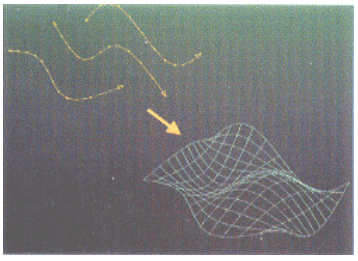


図 19　スキニング

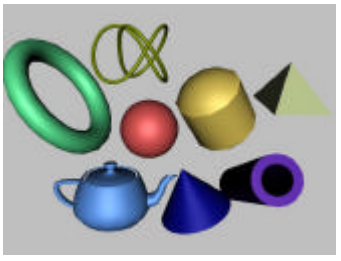


図 20 プリミティブ

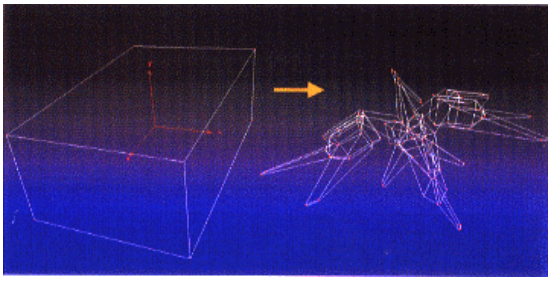
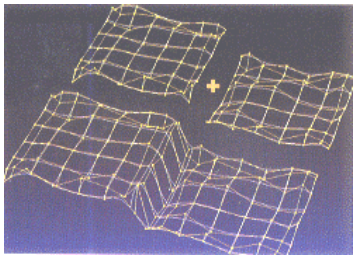


図 21　引っ張り出し



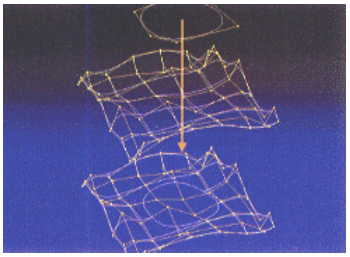
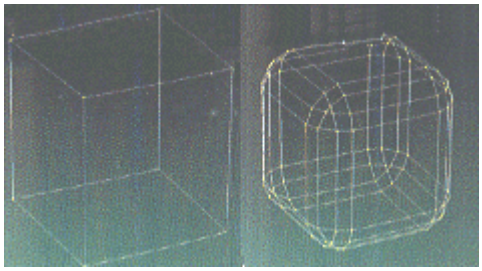


図 22－(a)　パッチのマージ　　図 22－(b) トリムカーブ



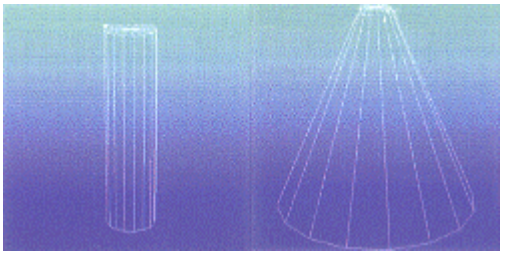


図 23 左がベベル前、右がベベル後　　図 24　左がテーパ前、右がテーパ後

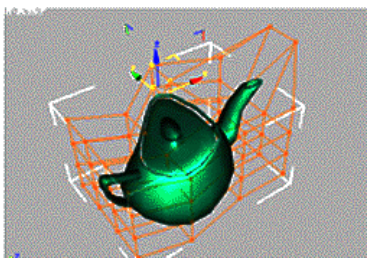
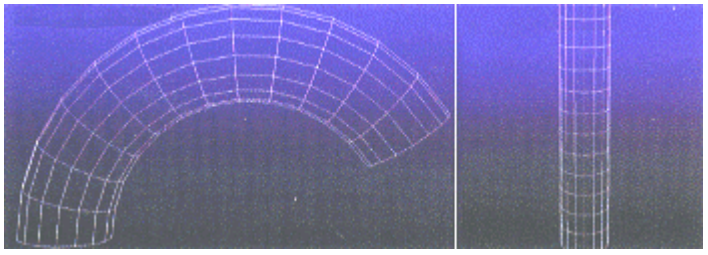


図 25　ラティス　　図 26　サーフェスフィット



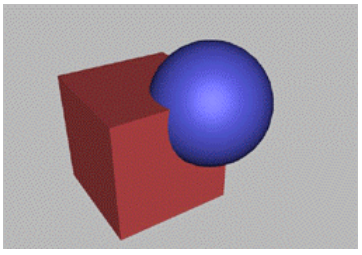


図 27　左がテーパ後　　右がテーパ後

図 28-(a) ブーリアン
立方体 A と球体 B

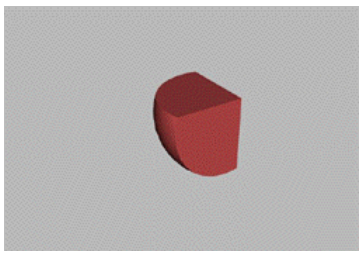


図 28-(a)　積　A and B

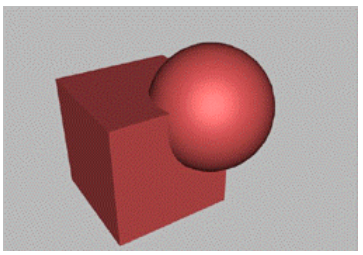


図 28-(a)　和　A or B

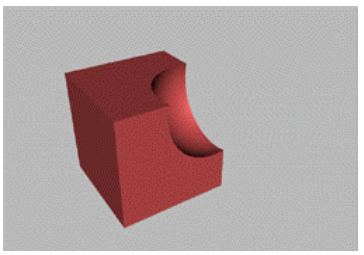


図 28-(a)　差　A－B

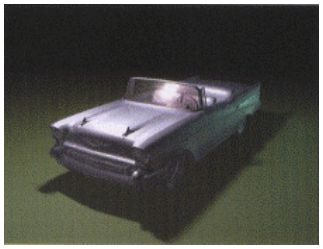


図 33　点光源

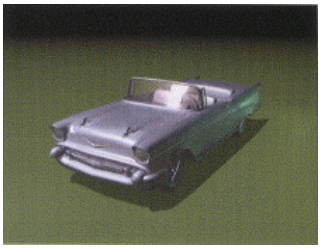


図 34　平行光源

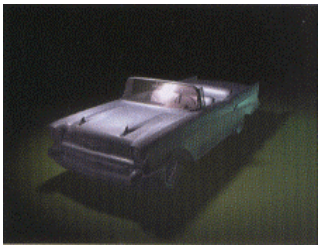


図 35 スポットライト

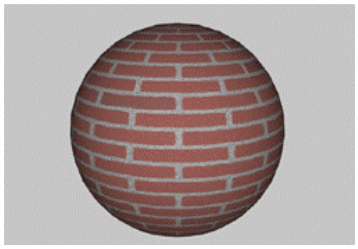


図 39 テクスチャマッピング

図 40 バンプマッピング

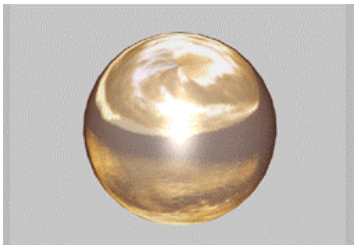


図 41　リフレクションマッピング

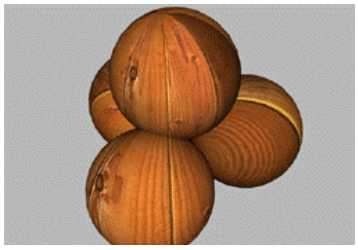


図 43 ソリッドテクスチャマッピング

図 44

図 45

図 46

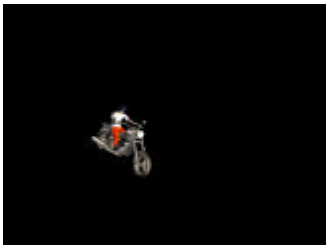


図 47

図 48

図 49

図 50

図 51

図 52

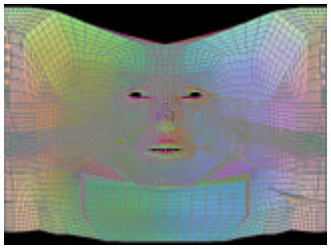


図 53

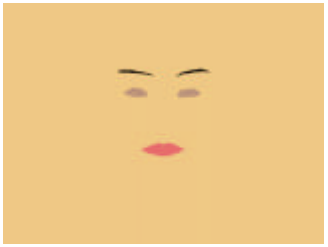


図 54

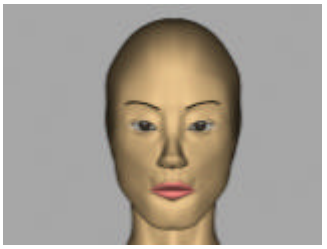


図 55

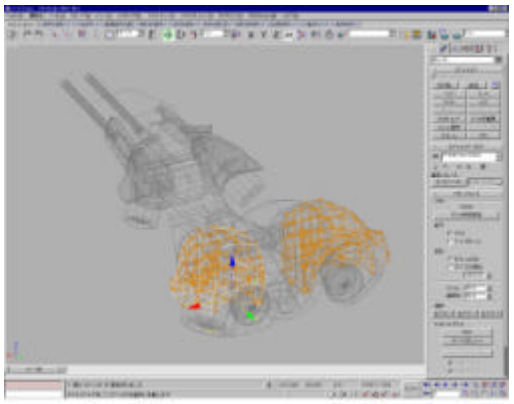


図 56

図 57

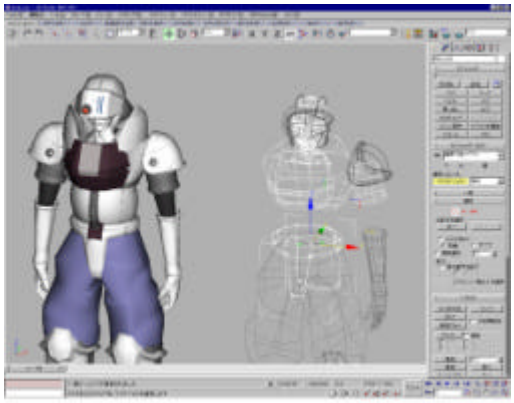


図 58

図 59

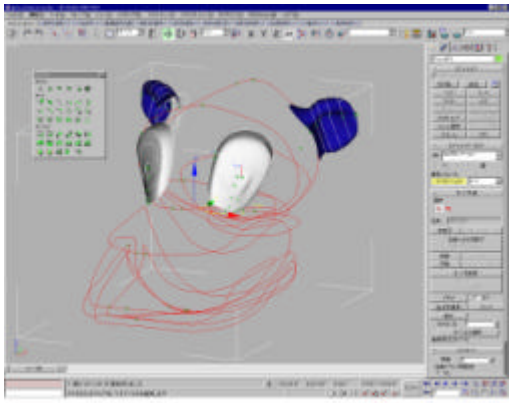


図 60

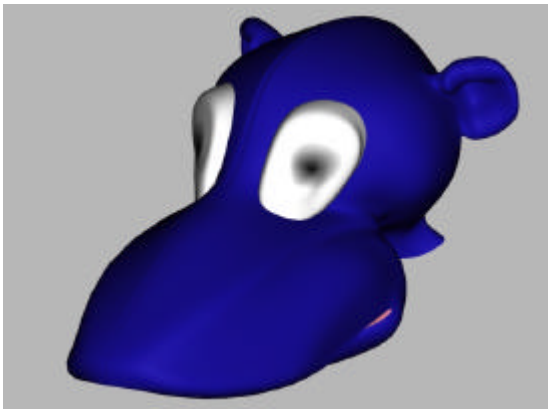


図 61

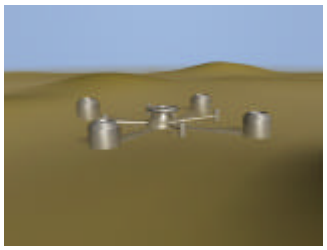


図 62

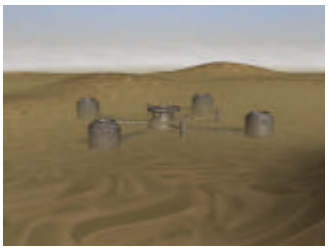


図 63

図 64

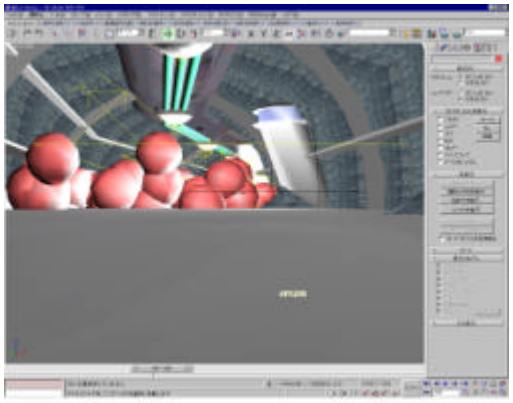


図 65

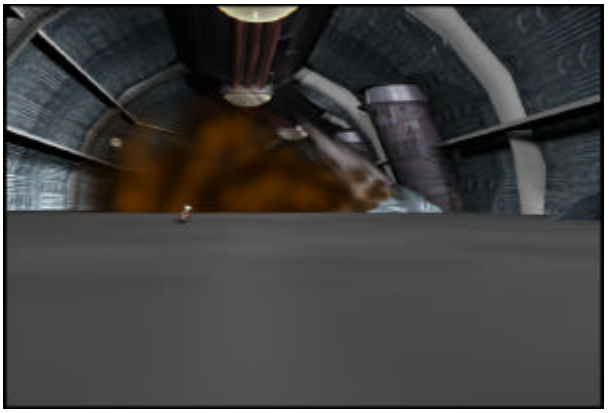


図 66

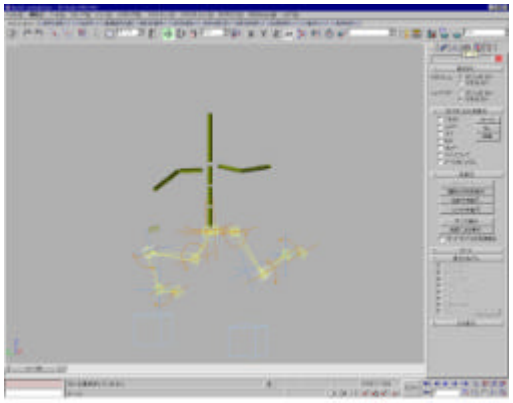


図 67

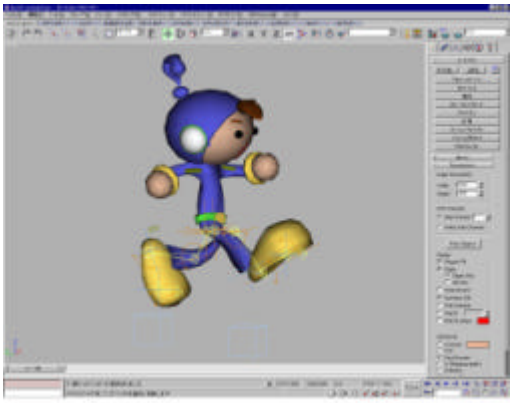


図 68

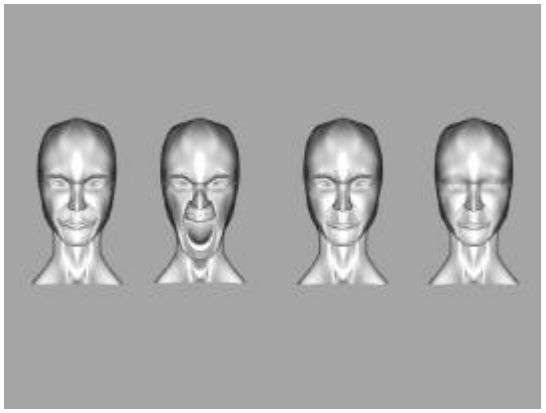


図 69

図 70