

2002 年度 卒 業 論 文

各種 3D グラフィックス最適化手法における  
複数の Web3D での差異に関する研究

指導教員：渡辺 大地

メディア学部 Web3D プログラミングプロジェクト

学籍番号 99P457

横川 隆史

2003年3月

2002 年度 卒 業 論 文 概 要

論文題目

各種 3D グラフィックス最適化手法における  
複数の Web3D 技術での差異に関する研究

メディア学部

学籍番号: 99P457

氏  
名

横川 隆史

主査

渡辺 大地

副査

金 尚泰

キーワード

Web3D 最適化 Web コンテンツ 3D モデル ファイルサイズ

Web3D を用いた Web コンテンツが増えてきている。商品紹介を主な目的として用いられているが、CAD などの工業用にデザインされたモデルをそのまま流用している場合も多く、家庭用パソコンでの処理速度やダウンロード時間など、Web 上で扱う場合に生じる問題を想定していない。閲覧者はノート型パソコンを利用しているなど、三次元描画能力が高くない環境にある場合や、通信に関して高速回線を利用していない場合が存在する。そのような閲覧者にとって、情報収集の際の Web3D コンテンツがストレスとなってしまうことは十分に考えられる。これらのパソコン性能、通信速度による問題はモデルのポリゴン数を減らすことで解決できると考えられるが、単純にポリゴンを削減するとモデルの描画品質は粗くなる。そこで 3D グラフィックス最適化手法を用いることによって、一般的な 3D グラフィックス同様、Web3D でもモデルの質感を保ったまま 3D モデルを表示できることに着目し、本論文では、複数の 3D グラフィックス最適化手法を複数の Web3D 技術に対して適用し、その結果それらの最適化手法が各 Web3D 技術に対して有効な手段として成り立つのか、またその結果から各種 Web3D 技術がどのような場合に有効となり得るのかを評価した。

## 目次

第 1 章	はじめに .....	1
1.1	Web3D の現状.....	1
1.2	Web3D における最適化の必要性 .....	2
第 2 章	各種 Web3D の評価 .....	4
2.1	VRML97 .....	4
2.2	Viewpoint.....	7
2.3	Cult3D .....	9
2.4	ShockWave3D.....	12
第 3 章	評価.....	16
3.1	Web3D 最適化の評価方法.....	16
3.2	描画.....	18
3.2.1	ライティング.....	19
3.2.2	シェーディング .....	24
3.2.3	マッピング .....	29
3.3	機能.....	32
3.3.1	最適化機能 .....	32
3.3.2	スクリプト .....	33
3.4	環境.....	35
3.4.1	動作環境.....	35
3.4.2	ファイルサイズ .....	37
第 4 章	考察.....	39
第 5 章	結論.....	44
謝辞	.....	47
参考文献	.....	48

# 第1章 はじめに

## 1.1 Web3D の現状

一般に Web3D とは、Internet Explorer や、Netscape といった Web ブラウザ上でインタラクティブに動作する 3D グラフィックスを示している。本論文でも Web 上で 3D グラフィックスを扱うこと、あるいはこの概念を Web3D とする。また、Web3D を表示する為の技術を Web3D 技術とし、Web3D 技術を用いた Web コンテンツを Web3D コンテンツ、Web3D コンテンツ内の 3D 画像部分について Web3D シーンと称する。

近年、Web コンテンツに Web3D 技術を用いて画像を表示するサイトが増えてきている。まだ広く一般的に用いられているというわけではないが、現状では主に携帯電話や自動車などの視覚的な魅力が売りに大きき影響する製品を扱ったメーカーのサイトで、より正確にあるいは面白く見せる目的で Web3D 技術を用いて商品紹介をしている場合が多い。わずかながら建築関係の会社でモデルルームのシミュレートに用いている場合や、個人で作った 3D グラフィックスを公開している場合もある。これら企業や個人が Web3D 技術を用いて商品などのアピールできるようになった背景には、グラフィックスに関するものを含めパソコンの性能が進歩し、数年前まではグラフィックスに特化したワークステーションでしか到底扱えなかったような 3D グラフィックスの表示に関して、家庭用デスクトップパソコンでもストレス無く 3D グラフィックスを扱えるようになってきたことが挙げられる。完成品としてではなく個人的にカスタマイズできるようにパソコンの部品、例えばメモリやグラフィックボードなどが個々として販売され、購入者各々のこだわりを反映したパソコンを作ることができるようになったこともグラフィック性能の向上に繋がっていると言える。

また、通信に関して近年をブロードバンド時代と称することがあるように、ADSL・光ファイバーに代表されるような低価格に設定された高速回線を家庭に引くことが可能になった。このことが単純な二次元画像よりもデータ量が多くなる 3D グラフィックスの送受信を現実的なものとし、Web3D コンテンツを扱う環境を整備されつつある。最低限の情報の送受信により成り立っているとはいえ、先のグラフィックスに関しての性能の向上と相まった効果として、3D 空間を共有したゲームなどが多数運用されている。3D グラフィックス表示機能と、ある程度以上のデータのやりとりが保証されていないとゲームとして成り立たないことから、数年前には到底一般ユーザ向きには考えられなかった環境が一般ユー

げに整えられてきているという証明になるだろう。

加えて、3D グラフィックスそのものの技術自体についても進化しており、それ程精細に3D モデルを完成させずとも、表現方法としてそのモデルをより精細に描画する手法や技術が開発されている。これらの表現方法を含め、様々な特徴を持った Web3D 技術が複数開発されている事自体も Web3D 技術の進歩につながり、Web3D コンテンツを表現手段として使いやすくしており、Web3D コンテンツを増やしている影響になっていると考えられる。

## 1.2 Web3D における最適化の必要性

様々な側面から Web3D 技術を扱える環境が作られているとはいえ、これらのグラフィックスや通信の環境改善が全てのユーザに整えられているわけではない。例えば、ノート型パソコンを利用しているユーザは数多く存在しているが、ノート型パソコンは3D グラフィックスを表示するのに適していない。ノート型パソコンの性能も上がっているが、デスクトップパソコンと比較するとその性能の水準はかなり低い。それでも小さく、軽い点を魅力としてノート型パソコンを利用しているユーザは多い。また、1.1 節においてオンラインゲームが人気を博していることを述べたが、ゲームに興味を示さない者もいる。こうしたユーザは3D グラフィックスをはじめとしたグラフィックスについてあまり大きな興味をもっていない可能性がある。使用しているパソコンのグラフィックス性能に多少の不足を感じているとしても、これを改善するだけのために、購入に十数万円は掛かる高価なデスクトップパソコンを購入するとは考えにくい。

また、回線速度に関しても、メールのやりとりや Web サイトの閲覧といった、短時間の接続を考えているユーザ、あるいは巨大なファイルのやりとりを必要としないユーザは、家庭に引く回線を選択する際、高速回線を引くメリットが無いと感じる可能性は高い。また、都市部から大きく外れてしまう地域では高速データに関する環境や設備が不十分で、地域の問題から高速回線が利用できないようなユーザも少なくない。

そのようなパソコンの利用環境の水準が高くはないユーザも、車や携帯電話の買い替えを始め、あらゆる商品の購入の為の下調べの為に各商品の製品紹介をしている Web サイトを閲覧する可能性は十分にある。その最中、偶然に Web3D コンテンツに行き着くことも考えられる。このようなユーザは、なかなか表示されない画像、表示されてもうまく動かな

い Web3D コンテンツに大きなストレスを感じる事が想定できる。

製品紹介や作品紹介に Web3D コンテンツを用いてより正確かつ魅力的に商品などをアピールすることは宣伝、主張の手法として大きな効果を持つと考えられるが、全てのユーザが同じ環境を持って Web を閲覧しているわけではないということを十分踏まえて Web3D コンテンツを作らなくては、ユーザに情報を与えるはずの Web コンテンツの意味が失われてしまう。従ってデータ量をより少なく、コンピュータへの付加はより小さくするといった工夫をこらす必要がある。だが、単純に 3D グラフィックスのデータ量を減らすと画像のリアリティが失われ、わざわざ 3D グラフィックスを表現手段として選択した意味が損なわれてしまうことになる。写真やイラストといった画像を用意せず、情報として不十分な 3D グラフィックスを用意したのでは、誰にとってもメリットが生じない無意味な Web コンテンツを制作することに他ならない。

しかし、Web3D は Web 上で扱うことを前提としているので、ポリゴン数が少なくても画像を綺麗に表示する機能や、インタラクションに関する機能を有している。それはそれぞれの Web3D 技術において、全てが同じ機能を持っているわけではない。本論文では 3D グラフィックスの画質を補う手法やコンピュータへの付加を軽減する手段を VRML・Viewpoint・Cult3D・ShockWave3D について適用していく。その結果からの考察として、どの Web3D 技術をどのような場面で用いればコンテンツをより上手く表現できるのかの提案をする。本論文において第 2 章では、各 Web3D 技術の表現力や機能面についての評価を示す。第 3 章では、各 Web3D コンテンツ用のモデルを最適化し、その結果や影響についての考察を述べる。第 4 章では、第 3 章での評価結果を考察し、第 5 章にてどのような場面で、どの Web3D 技術を用いるのが適切なのかを提案する。

## 第2章 各種 Web3D の評価

Web3D コンテンツを閲覧する際、Web ブラウザ自体にはインタラクティブな 3D グラフィックスを表示する機能は備わっていないので、Web3D 技術毎に専用のビューワーを必要とする。HTML 文書の記述中に、Web3D シーンを呼び出すタグを挿入することで、その該当箇所に Web3D ビューワーと 3D モデルまたは 3D 空間が呼び出され、Web コンテンツ中に 3D モデルが挿入される。Web3D 技術はただのリアルな 3D 画像を表示する技術ではなく、3D モデルの回転や移動などの多くのインタラクティブ機能を備えており、Web コンテンツのインタラクティブ性を向上させる効果を持っている。このインタラクティブ機能は Web3D 技術によって実装されているものに差異があり、これが各種 Web3D 技術それぞれの持つ特徴となる。最適化の評価の前に本論文では Web3D 技術の中から VRML、Viewpoint、Cult3D、ShockWave3D についての紹介と評価をする。

### 2.1 VRML97

数ある Web3D 技術の中で最も古いものが VRML である。現在は VRML2、または VRML97 と呼ばれる形式が主に扱われている。パソコンのあらゆる環境において、Web3D に限らず 3D グラフィックスを扱うには未熟だった頃に作られたものである。最初の Web3D 技術であるため、他の新しい Web3D 技術と比較すると、ほとんどの機能や表現力で劣っている。見た目に最も影響するシェーディングについて他の Web3D 技術との差が顕著に表れて見られるが、3D モデルの表面を滑らかにする手法であるスムーズが一切かからない、フラットシェーディングによってレンダリングされる。VRML で書き出したモデルを図 1 に示す。ポリゴンのエッジがそのまま表示されてしまうので、簡素なモデルを作ってしまうと曲面を上手く表現することができない。モデリングツールのスムーズ機能を使ったままモデルを作成すると、VRML で出力されたものが想像していたものよりもはるかに粗悪なモデルとなる。

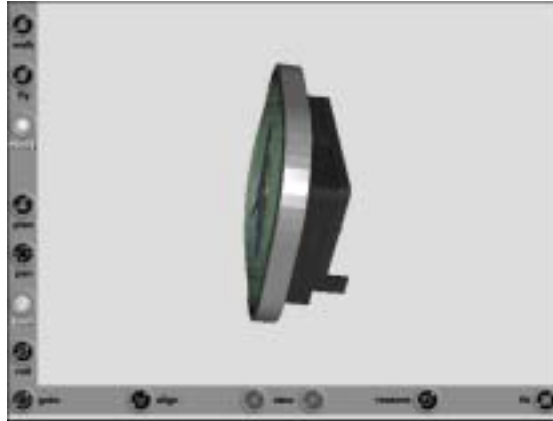


図 1 VRML

VRML のスクリプトを記述したファイルは基本的には圧縮されない。ビューワーによっては gzip 形式で圧縮したものを読み込めるが、全てのビューワーで読み込めるものではない。立方体や球体といった単純な形状は“ Box ”、“ Sphere ”といった記述で呼び出すことができるが、複雑な形状となると、そのモデルの頂点 1 つ 1 つの座標を数字で書き込む必要があるため、モデル全体が完成する頃には Web コンテンツの一環として扱うには適さない程巨大なファイルサイズとなってしまふ。今回、最適化に使用する素材を用意するに当たり東京工科大学の AQUA-PROJECT から借用したハナミノカサゴ(以下、魚)のファイルサイズについては 2MB で、ポリゴン数を削減しても 1MB までしかファイルサイズは減少できなかった。1MB のファイルは、Web コンテンツに用いる画像としてはかなり巨大である。このファイルの内容は圧縮されないので編集しやすく、既存の作品を参考にして VRML の手法を学ぶことが容易なのだが、全ての頂点を記述しなければならない点など、効率的でないスクリプトが痛手となるようだ。また、工業デザインを保守したい企業にとっては、デザイン情報が正確な数値として流出してしまうこのファイルの形態は受け入れ難いものだと考えられる。

様々な不足がうかがえる VRML だが、表現される 3D モデルはただの静止画像になることはない。他の Web3D 技術と同様に、Web コンテンツでインタラクティブに操作できるようにする機能を備えており、少なからず Web 上で扱う点について配慮されている。モデルにマウスでの操作可能な属性を与え、マウスクリックによって音楽やアニメーションを操作できるようにする、あるいはモデルにハイパーリンクを貼って別サイトにジャンプすることも可能である。ただ、最も基本的なインタラクションと言えるアニメーションに関



してデフォルトでは自動的にループしてしまう、マウスによる操作や、オブジェクトとカメラの距離などである程度アニメーションの制限できるが、その場合クリエイターの思い通りのタイミングでユーザが操作するとは限らないといった問題がある。思い通りのアニメーションを付ける為にはインタラクティブ性を無視して全て自動で動くように作ってしまうことになる。これでは Web3D コンテンツというよりはただの動画になってしまう。

視点の操作に関しては多くの Web3D 技術の中でも最も自由に動かすことができるといえる。Web ブラウザに組み込まれた VRML ビューワーの一部は、3D モデルを表示せずカメラの操作パネルとして表示される。ズーム、回転、パンについてそれぞれにボタンがあるため、操作に必要な手続きが多く、初心者には慣れが必要で操作に戸惑いを感じると考えられるが、回転についてカメラの座標を中心に回す状態と、オブジェクトまたは 3D 空間の中心を回す状態との区別ができるなど、移動方法が非常に豊富で思い通りに動かせる。表現力に難があるとはいえ、カメラの挙動自体はオブジェクト単体でも 3D 空間の移動でもどちらにも対応できる。デフォルトではカメラの動きは遅すぎるが、右クリックから開かれるビューワーのカスタマイズの項目中にカメラに関するものがあるので、そこでカメラの速度を変更できる。また、VRML はビューワーが複数あるのも特徴である。アンチエイリアスなど基本的な機能についてはほとんどのビューワーでサポートされているが、それ以外のビューワーの機能が他のユーザも同様に使えるとは限らない。ユーザのビューワーを限定することができないため、ビューワーの機能に頼って画像の表現効果を上げる、ということはあまり適切な手段ではないと言える。

比較対象となる他の Web3D 技術がなければ活用方法は幅広いのだが、前述の通り他の Web3D 技術と比較すると問題となる部分が多い。それでも、3D 空間を移動するマンションのモデルルームの紹介には十分に使えると考えられる。この場合、多彩なカメラアクションで自在に空間移動ができる点が大きなメリットとなる。また、部屋の中にある家具などは直線のみで表現できるモデルが多いので、モデルのエッジが目立つ事が少なく、違和感なく表現できる。3.3.1 節で説明するが、LOD 機能を備えているので、3D 空間を扱う時にはコンピュータへの負荷も小さくできる可能性がある。VRML に関しては[1][2][3]など、多数の情報源が存在する。

## 2.2 Viewpoint

3D モデルの滑らかな表現を得意としているのが Viewpoint である。特別な処置をとらなくても、ポリゴンのエッジは見えなくなるようにシェーディングされる。その画像を図 2 に示す。鋭角的な境界も滑らかに表現されてしまうこともあるが、エッジの表現についてポリゴンのエッジを黒い線で描かせることが可能である。エッジを描いたものを図 3 に示す。エッジを全て描くことも可能だが、輪郭線のみを描くこともできる。輪郭線について 2 種類の輪郭の選別方法があるので、クリエイターの求める方を選択すればよい。滑らかな画質を実現する点で、Viewpoint と Macromedia の Flash の両技術を同一コンテンツ内で扱った場合、視覚的な整合性が高い。技術的にも Flash から Viewpoint を操作することが実現され、両技術の連携・融合が積極的に図られている。



図 2 Viewpoint



図 3 Viewpoint 縁取りつき

Viewpoint の照明には大きな特徴がある。照明が他の Web3D 技術よりも強く照らされ、柔らかくに広がっている。この照明が光の反射を綺麗に見せ、表面の滑らかさと相まって 3D モデルを非常にリアルな画像に見せる。照明に対する影も絶妙で、視点から離れた部分、モデルの下の方が滑らかに暗くなっていく、ソフトドロップシャドウと呼ばれる技術を使っている。金属、ガラス、プラスチックのような表面の光の反射が大きな材質を表現するのが得意で、照明による反射が実物どおりの反射ではなくても、光の具合、影の具合が感覚的にリアルだと感じさせてくれる。モデル単体を見せる場合における効果が非常に大きい照明だが、3D 空間の表現となると具合が変わってくる。視点から遠くにあるものでも

同じように明るく見えてしまい、手前にあるモデルの影と、その奥にあるモデルの明るい部分の明度差が大きくなり不自然に見える。これはモデリング中に照明を追加する、あるいは Viewpoint ファイル中に照明を追加する編集を加えることで回避できるが、Viewpoint が本来持っている表現力を壊してしまうことになりやすい。また、パーティクルアニメーションを扱えない点、画像を表示する際に画面外のモデルに対しても常に処理をしている点からも、空間表現やシミュレートなど、多くのモデルを一度に処理することには向かないことがわかる。物体単体を示す場合は、ただモデルを示すだけでなく、Z-Buffer やテキストの表示などの機能の付加が可能で、図解などにも用いることができる。これらの機能の付加をする場合には、ファイル内容を編集する必要があるが、その際 Viewpoint Scene Builder を使用することでファイルの記述の仕方がわからない者でも作業できる。Viewpoint Scene Builder の画面を図 4 に示す。また、Viewpoint による Web3D コンテンツの制作方法は [4][5][6][7][8]に記述されている。



図 4 Viewpoint Scene Builder

アニメーションに関しては、VRML のように常に動きつづけてしまうということはなく、

アニメーションの再生回数の設定ができる。1 周のみの再生はもちろん、2 周、3 周アニメーションを繰り返すといった設定も可能である。Java Script による HTML 部分からの操作も可能で、任意のタイミングで再生・停止できるボタンをつけることもできる。この際、アニメーションのフレーム中でアニメーションのどの部分を使うか決められるので、複数のボタンを用意することで任意のアニメーションをコントロールさせることができる。

Viewpoint ファイルは 2 つ用意される。1 つは 3D モデルのデータ[\* .mts]である。ファイルの内容は圧縮され読むことができない。Viewpoint Stream Tuning Studio というアプリケーションのみ、ポリゴンやテクスチャについて編集することができる。モデルの形状に関するデータは Stream Tuning Studio でもできないので、モデルのデザインについて、正確な情報が漏れてしまうことが無い。もう 1 つはモデルの配置に関するファイル[\* .MTX]で、モデルの大きさやアニメーションなどが記述される。このファイルはテキスト形式で記述されるので Notepad などでも編集が可能である。mts ファイルについては非常に小さなサイズに圧縮され、MTX ファイルは非圧縮ファイルだが最小限の記述で済むように設計されている。Web 上で扱うことを考え、ファイルサイズを小さくとどめようとする配慮がうかがえる。VRML で 2MB を必要とした魚のファイルだが、Viewpoint ではわずか 153KB、モデルにもよって差が出るが、VRML の 100 ~ 1000 分の 1 のファイルサイズで同程度のデータ品質を実現できる。Viewpoint ではテクスチャの画像についても独自の圧縮をすることが可能で、画像に関してもサイズを大幅に削ることができる。

Viewpoint は有償の技術で、ブロードキャストキーを購入しないと、オンラインでファイルを開いたときに画面中に「Viewpoint」と書かれたレイヤーが描かれ、細部が見えなくなってしまう。利用者の利益によって使用料は変わり、利用目的が非商用の場合ブロードキャストキーは無料で提供される。また Viewpoint での表現・製作に必要なツールは全て無償である。

## 2.3 Cult3D

Cult3D は物体単体、3D 空間、シミュレーションいずれにも対応している。モデルの質感は VRML の技術を向上させたものといったところで、シェーディングの方法を選択することで VRML 同様のもの、それよりも遥かにリアルなものを表現できる。Cult3D で書き

出したモデルを図 5 に示す。

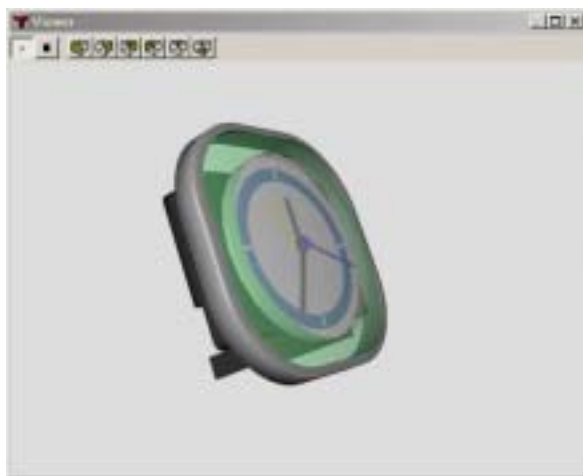


図 5 Cult3D

Cult3D は Cult3D Designer というソフトで編集される。モデルの回転をとっても、どのオブジェクトを、何を中心に回すのか、といった細かな設定が可能である。編集作業も簡単にイベントマップと呼ばれる空のウィンドウに、設定を適用するタイミング、アニメーションなどのアクション、それを適用するノード（オブジェクト）を別ウィンドウからドラッグするだけである。メニューからそれぞれを関連付けすることもできるが、イベントマップ上のアイコンにドロップすれば自動で関連付けしてくれる。細かな設定はそれらアイコンから開かれるメニューで編集できる。Cult3D Designer の画面を図 6 に示す。編集できる項目は多いがマニュアルに各項目についての詳細が書かれていない、Cult3D Designer のプレビューと Web 上ではマウス操作に対する反応が違うなどの影響で Cult3D Designer で思うように設定できないということが少なくない。いずれの不都合も慣れてしまえば問題は無くなる。

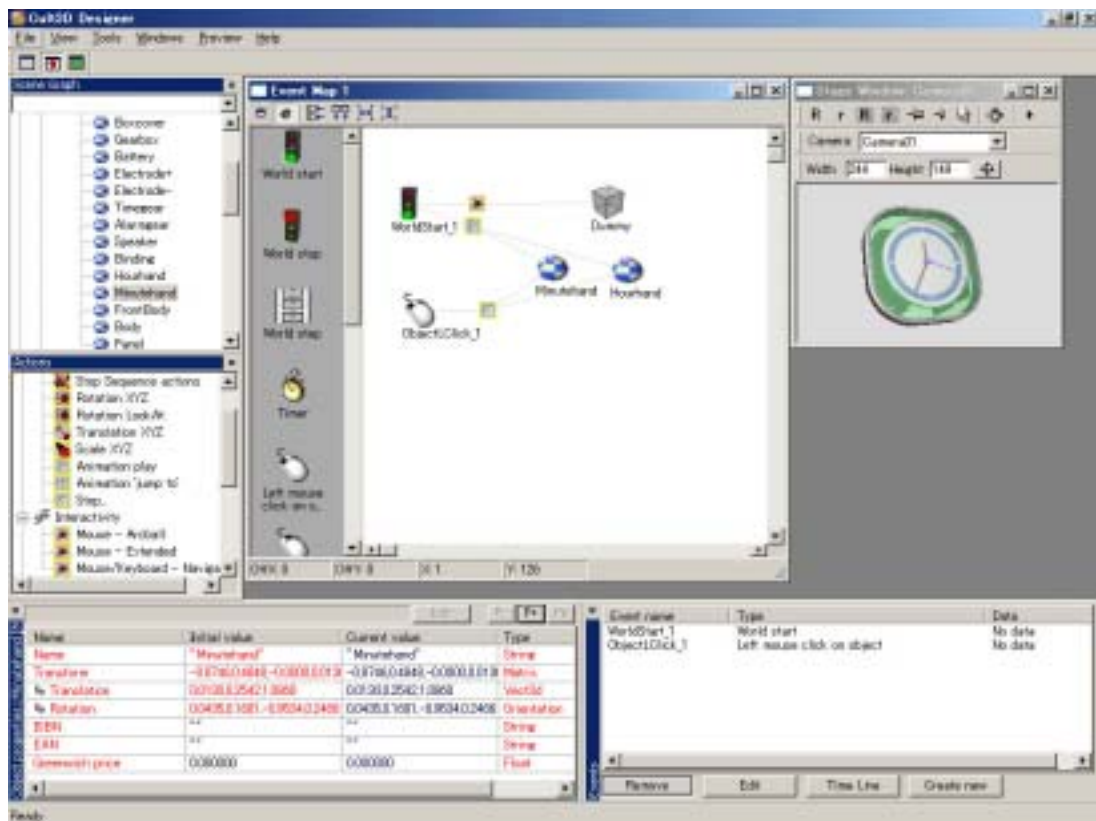


図 6 Cult3D Designer

Cult3D Designer ではオブジェクトのアクションに関して、13 項目のカメラやアニメーションなどのセクションがあり、全部で 40 のアクションが用意されている。さらにそれらについて、そのアクションの挙動についての設定をする。また、Java の Class ファイルの読み込みで多彩な動作を実現できる。他の Web3D 技術よりもアクション定義の機能が充実している。

アニメーションに関しては Viewpoint 同様の操作が可能である。キーボードからの操作も受け付けるので、より多くのアクションがつけられる。また、パーティクルアニメーションの利用が可能で、水の流れや爆発などを含めたシミュレーションも可能である。Web3D 技術では珍しく、パーティックスアニメーションにも対応しており、メッシュの形状の変化するアニメーションも表現できる。具体的には、人が歩くようなアニメーションの場合に、関節付近のポリゴンのメッシュが伸び縮みするので関節付近が非常に滑らかに表現される。通常の Web3D 技術ではこのようにメッシュ形状が変化するアニメーションは扱えない。

Cult3D のファイルは、モデリングソフトから書き出された時にできる[\*.\*c3d]ファイル、

c3d ファイルを Cult3D Designer で編集したものが保存される[\* .c3p]ファイル、Web 公開用の[\* .co]ファイルの 3 つがあり、Web 上では co ファイルのみを公開すればよい。この co ファイルは Cult3D Designer でも編集することができないので、デザインが外部に漏れることがない。co ファイルは書き出された後には絶対に編集されないで、非常に高い圧縮率で圧縮されている。VRML の魚 2MB に対して、631KB とこちらも非常に小さなサイズに収まっている。

非常に多くの操作が可能で、画像も綺麗な Cult3D だが、他の Web3D 技術よりも処理が重くなりがちである。VRML よりも滑らかなシェーディングが 2 種類用意されているのだが、そのいずれとも、3D 空間上に多数のオブジェクトを用意すると、三次元描画能力の高いグラフィックボードを積んでいないようなパソコンでは、快適に閲覧することができない。

Cult 3D も有償技術である。Cult3D シーン生成に必要な Cult3D Designer のシリアルキーを購入していない場合、購入を促すメッセージがレイヤーに描かれる。使用料は使用者のサイトのアクセス数によって異なり、非営利目的の使用の場合のみ無料でシリアルキーを得ることができる。

## 2.4 ShockWave3D

ShockWave3D は Web 技術として馴染みの Flash を提供している Macromedia によって開発されている。モデルの質感については有益な特徴はあまりなく、3D グラフィックスが持つリアルな光沢や質感は表現されない。ShockWave3D のモデルを図 7 に示す。

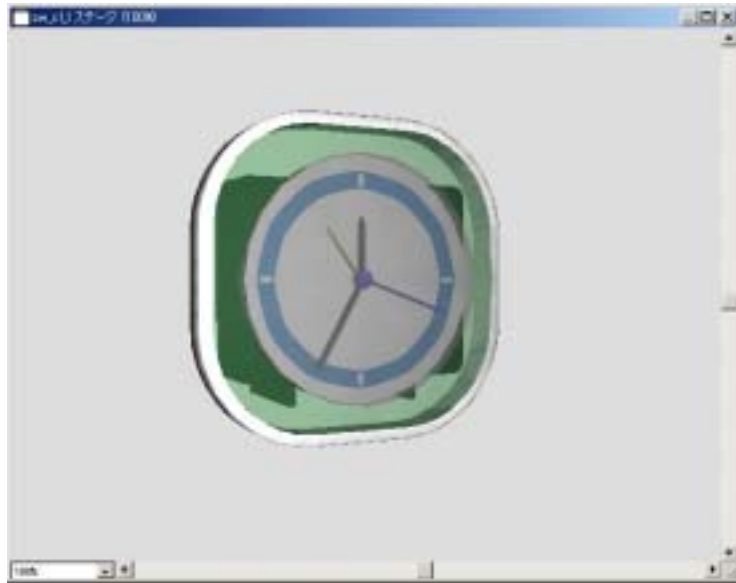


図 7 ShockWave3D

光が強く当たる部分はその周囲よりも明るくなる程度で画像として無難な印象がある。VRML ほど無骨な画像でもなく、特別綺麗なモデルになるわけでもない。しかし、ShockWave3D は Flash 同様に Web コンテンツのインタラクティブ性を向上させる効果が大きい。他の技術で Java を用いて機能の拡張するのに対して、ShockWave3D では Lingo という独自の言語での Web3D コンテンツの機能の拡張が可能である。他の Web3D 技術では、その技術が標準で備えられている以上の機能拡張したい場合に Java のクラスファイルを読み込まなくてはならないが、ShockWave3D では Lingo でプログラムしたことがそのまま反映される。Lingo で記述や処理することができないことは ShockWave3D で表現することもできないと明確である。この Lingo は ShockWave3D の見た目や機能に対する機能拡張だけでなく、プログラム言語として十分に使うことができる。そのため、他の技術では扱えないようなプログラムを ShockWave3D のコンテンツの中に組み込んでしまうことができる。Flash 同様に Web3D を用いたオンラインゲームやその他のインタラクションなど、多数の利用方法がある。また、ShockWave3D はその 3D 空間を複数のユーザで共有することができ、ゲームや 3D キャラクタを用いたコミュニケーションサイトの表現技術として用いることが可能である。

ShockWave3D は Director で編集することになる。Director の画面を図 8 に示す。モデルの回転やカメラの動作など基本的な制御に関するスクリプトはあらかじめ用意されて



おり、ドラッグアンドドロップの操作で実現できる。これらのすべてが Lingo で記述されていて、スクリプトウィンドウを開けば各自の望みどおりに編集することができる。また、Lingo がただのエディット用のスクリプトでなく、プログラム言語として通用する性質を持っているので、モデルに重力の影響を与えるなど物理シミュレーションも行える。パーティクルアニメーションについては、外部のモデリングソフトから書き出さなくても、Lingo が機能を備えているので簡単に実現できる。

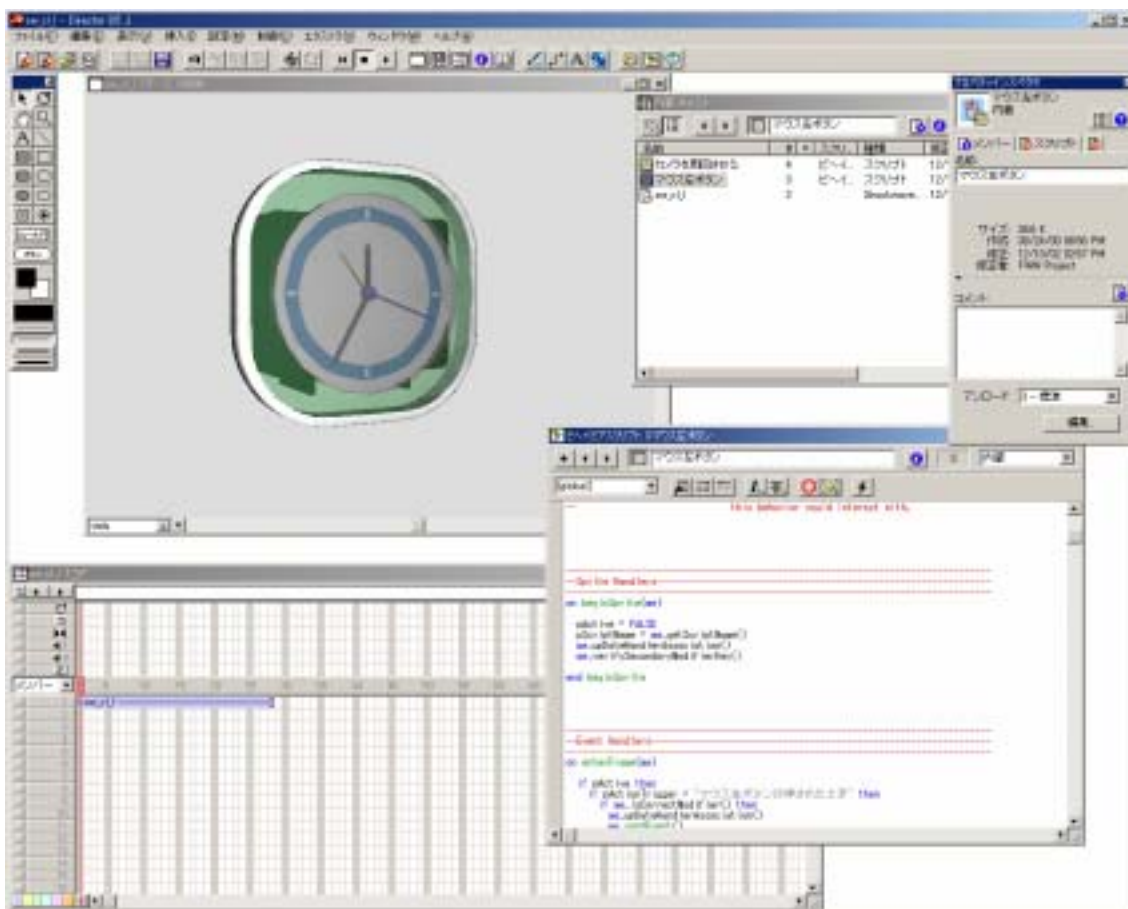


図 8 Director

単純にモデルを表現する Web3D コンテンツには、他の Web3D 技術の方が勝ると言う意味で ShockWave3D は不向きで、プログラムとの連携を得意としている点から、3D 空間の表現やシミュレーション、さらに Web 上のアプリケーションとしての表現に向いている。事実、ShockWave3D に関する書籍[9][10][11]では ShockWave3D を用いてオンラインゲームを作る項目が見られる。

同社の製品である Flash については、2.2 節に示したように Viewpoint との融合を図っている。そのため ShockWave3D では Viewpoint に乏しい機能面についての技術が進んでいるようである。ファイルは Cult3D 同様に、モデリングツールから書き出される W3D ファイル、保存用の Dir ファイル、Web 公開用の dcr ファイルの 3 つに分けられる。

ShockWave3D に使用料は求められないが、ShockWave3D シーンの生成のために Director を購入する必要がある。

## 第3章 評価

### 3.1 Web3D 最適化の評価方法

最適化の評価対象として、意図の違う2つの3Dモデルを用意した。用意したモデルを図9に示す。1つは無機質なものとして時計を用意した。形が単純でテクスチャを用いずとも物質の表面の色（以下マテリアル）によって、物体を表現できるものである。単純なモデルであるために、最適化を施した場合にその影響が出やすい。もう1つ自然物として魚を用意した。これは2.1節にあるもの同様、AQUA-PROJECTから借用したモデルである。複雑な曲面とテクスチャによって表現され、ポリゴン削減ツールで大幅にポリゴンを削減できる。自然物としての質感は損なわれるが、テクスチャマッピングの影響を評価するのに適している。また、テクスチャの貼られていないものも用意し、幾何学的なデザインの時計と自然物（魚）との差が見られるようにした。それぞれにポリゴン数が多く滑らかに見えるものと、ポリゴン数が少なく粗雑に見えるものを用意した。後者に対して3Dグラフィックス最適化手法を流用することで、前者との見た目の差がどれほど縮まるのか、各種Web3D技術がどのような最適化機能を有しているのかを評価する。

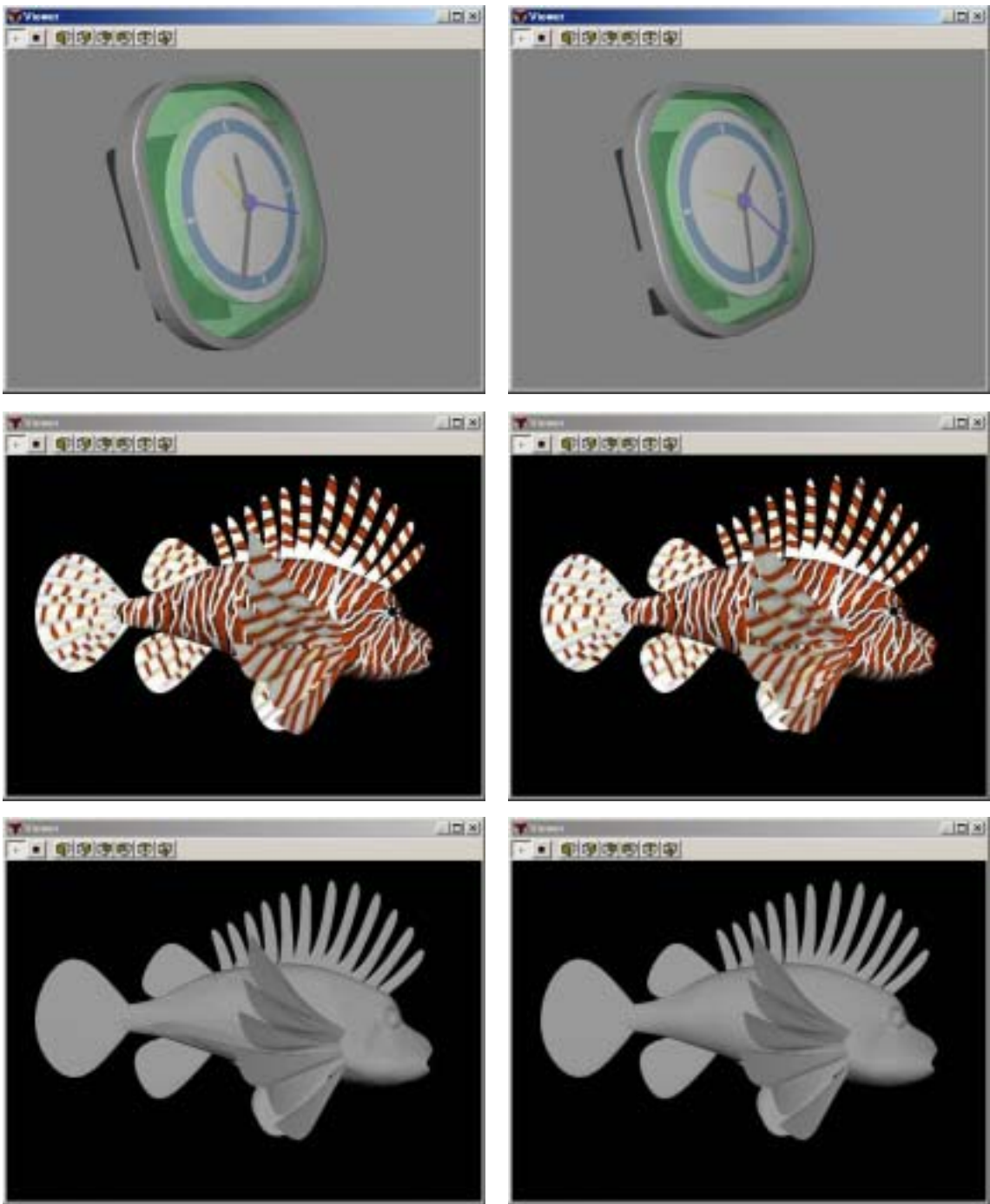


図 9 用意したモデル

左がポリゴン数を落としたモデル、右がポリゴン数の多いままのモデル

Web3D 技術についての評価対象は、描画・機能・環境について大別されると考えられる。

描画は、各 Web3D 技術によって画面上に書き出される画像に関わる項目で、照明、シェーディング、マッピングに関しての評価をする。照明に関して、照明を和らげるなどの手段でモデルの表面の質感を変えることや、ポリゴンのエッジを見えにくくすることが可能である。シェーディングに関して、モデルの表面の拡散光の計算方法の違いからモデルの粗さを消すことができる。これは、各 Web3D 技術がどのような方法でシェーディングできるのか、それらを選択することが可能なのかをまとめる。マッピングについて、表面に模様がつくことで表面の粗さは目立たなくなる。また、バンプマップを適用可能ならば、テクスチャによってモデルの凹凸を表現できることになる。これら描画項目に関して 3D グラフィックスに関する書籍を参考にした[12][13][14]。

機能は、各 Web3D 技術がもつ描画以外の機能に関しての項目になる。各 Web3D 技術がどのような最適化機能を有しているのか、編集可能なスクリプトで書き出される場合、スクリプトをインスタンス化することなどが可能か、またはそのファイルを圧縮できるか、以上の点について示した。

環境は、本論文では単純な数値に関しての項目である。各技術の最低動作環境・推奨動作環境について、同じモデルのファイルサイズについてである。各 Web3D 技術ではファイルサイズの違いが生じるだろうと直感的に予想できる。また、そのファイルの予想ダウンロード時間も通信速度ごとに示す。これらの項目に関してはモデルの状態と、ユーザの環境に大きく影響されると考えられるため、簡単な目安程度のものとして示す。

## 3.2 描画

ライティングについては、照明にぼかしをつけることで表面のエッジの目立ちを抑えた結果を示す。また、それによる弊害が生まれる場合について考察してある。シェーディングは、Web3D 技術の機能と大きく関わってくる部分だが、見た目への影響が非常に大きいことから本項目に加えた。各 Web3D 技術がサポートしているシェーディングの種類とそれによる効果とについて考察している。高度なシェーディングを適用した場合、処理が重くなるなどの害があるが、そもそも Web3D 技術が備えている機能ということで、大きな問題としては捕らえていない。マッピングに関しては、適用可能な画像、またその効果について考察している。

### 3.2.1 ライティング

照明の効果によるモデル表面の質感の向上だが、単純に照明を加えた時点で効果が出るわけではない。本論文で取り上げた4つの技術中、Viewpointを除いた技術に関して、モデリングツール、あるいはWeb3Dシーンの編集作業中に照明を加えた時点でデフォルトの照明は削除されてしまう。その為、最適化として照明を追加した面以外にも、モデルを十分に表現できるだけの照明が必要とされた。最低でも初期視点から見て手前、奥、左、右、上、下の6つが必要で、モデルの形状によって、照明の配置の角度などを調節する必要がある。もっともファイルサイズが大きくなるVRMLについて、3DstudioMaxにおけるオムニライト（照明の方向性が無く、電球のように画面全体に広がっていく光）1つにつき145バイトが必要とされた。また照明の個数について、より複雑な形状を持つモデルについてはこの6つ限りではなく、モデル全体を不足なく照らせるだけの個数が必要となる。VRMLにおける結果を図10、図11示す。

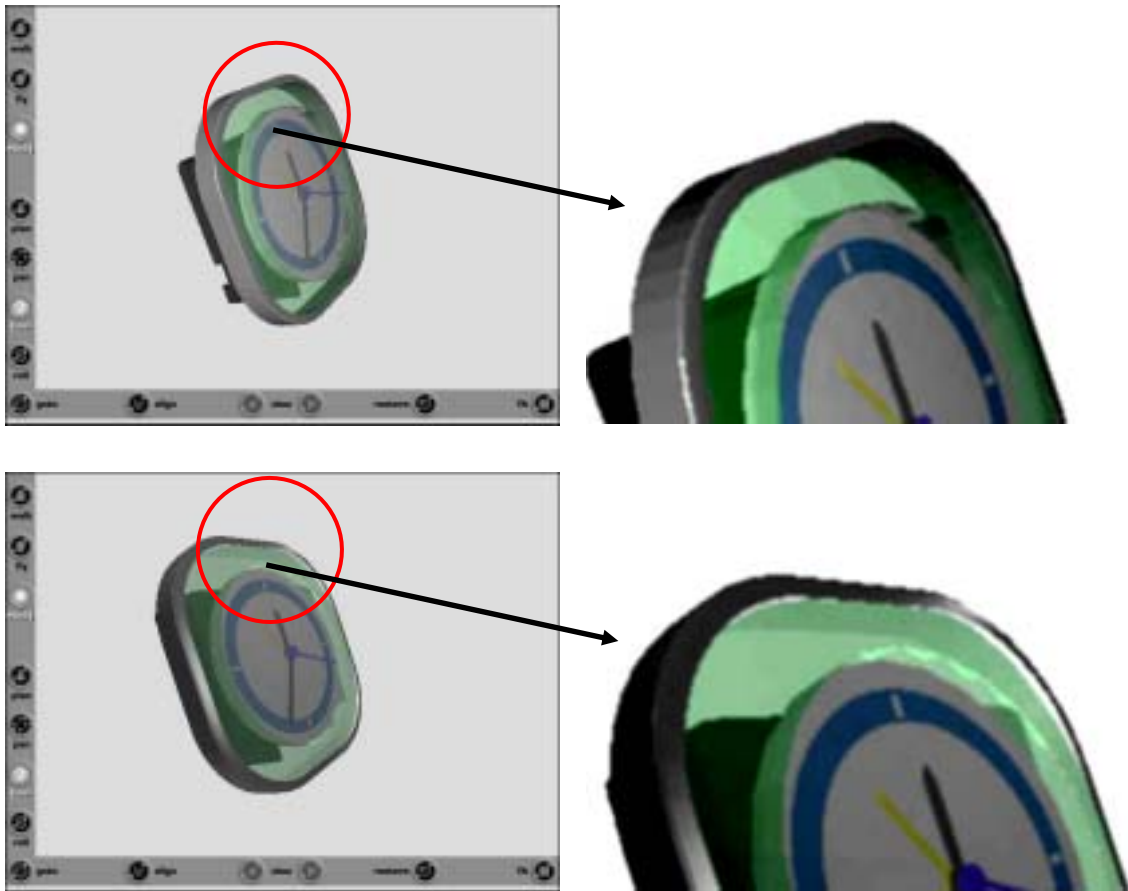


図 10 VRML と Cult3D における照明による最適化（画像は VRML）  
上が照明追加前、下が照明追加後。時計では正面や枠のエッジが見えにくくなった

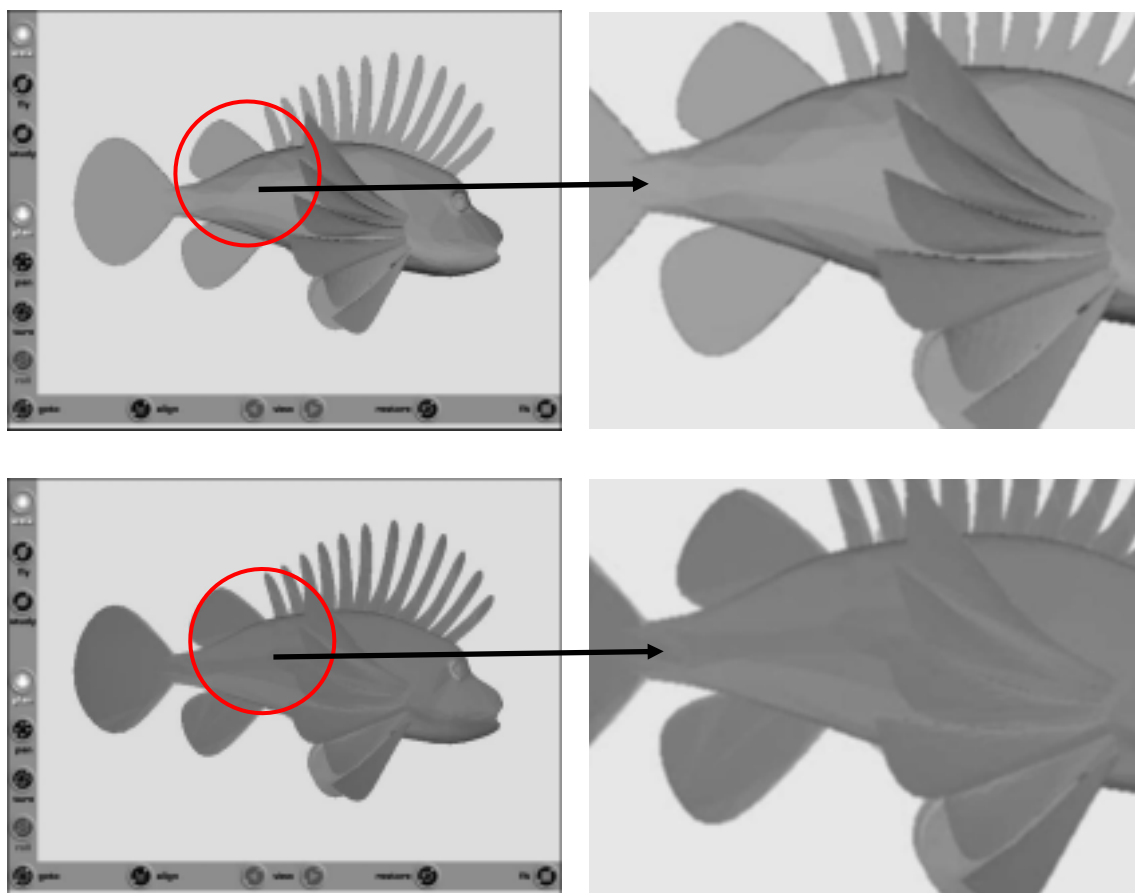


図 11 VRML と Cult3D における照明による最適化（画像は VRML）

上が照明追加前、下が照明追加後。魚では輪郭が滑らかに描かれたように見える

時計についてはライト 6 つ、魚についてはライト 8 個を備えている。両者のモデルについて、追加した照明がデフォルトの照明よりも強い光を発してしまうので、照明の輝度を調節してモデルが眩しくなりすぎないように調節が必要とされた。デフォルトの照明はカメラの操作と一緒に動くが、追加した照明は 3D モデルと一緒に動く。そのため、常に強く光の当たる部分がでてしまった。照明を弱くし、位置について時計に照明が真正面に当たるのではなく、角に当たるように調節し、物体の角は光って見えるという感覚的な印象に合わせて違和感を小さくした。テクスチャを貼った魚についてはまったく差があるように見えなかった。そこで魚のテクスチャを剥がし、魚についてどのような変化が実際に起きているのかを見た。テクスチャのない魚に関しては、金属部分がないので反射光に違和



感が出るという害はなかったが、急角度で接している面について明るい面と暗い面が生じてしまい、照明が強い場合に逆にエッジが目立ってしまった。明るくなっている面が照明を追加する前よりも明るくなっていることが見受けられたので、照明の輝度を落としたところ、面のコントラスト差が小さくなっていった。モデルの素材の明度と照明の輝度を同じ値にしたときが、エッジが目立たず、モデルも暗くなりすぎないという結果が出た。

**Viewpoint** についてはモデリングツールでライトを追加してもデフォルトの照明が失われることはなく、その為モデル全体を照らせるようにカメラを複数用意する必要がなかった。しかし、**Viewpoint** での画像はエッジが際立って見えることがほとんどなく、照明でエッジを消すという作業自体が必要ない。照明についての調査で、何らかの都合で照明の色や強さを変えたい場合は、手続き型ライトマップを使う必要があることがわかった。デフォルトの照明を消す必要がある場合、これによって照明が上書きされる。

**Cult3D** では **VRML** 同様の変化があった。**VRML** に比べ、ファイルサイズが小さくまとまるので、照明分のファイルサイズの増加が気にならない。本項目では、凹凸が露骨に表現されるフラットシェーディングにおいて照明の効果を検証したが、次項目で評価するシェーディングによる効果の方が大きく、他のシェーディングを用いた場合は照明を使用して質感を補う必要を感じられないので、照明による最適化は現実的には考えなくて良い方法と言える。**Viewpoint** 同様、最適化を意図とせず、**Web3D** コンテンツに効果を加える意図で照明を追加する場面の方が多いだろう。

**ShockWave3D** では、デフォルトの状態ではモデル正面に 1 つしか照明がなく、モデルと一緒に回転してしまう。その状態を図 12 に示す。**Director** 上で照明の位置を変えることは可能だが、モデルとともに照明が動くことに変わりはない。何らかの手を打たないとモデルの裏側には照明が当たらず、見ない状態のままである。

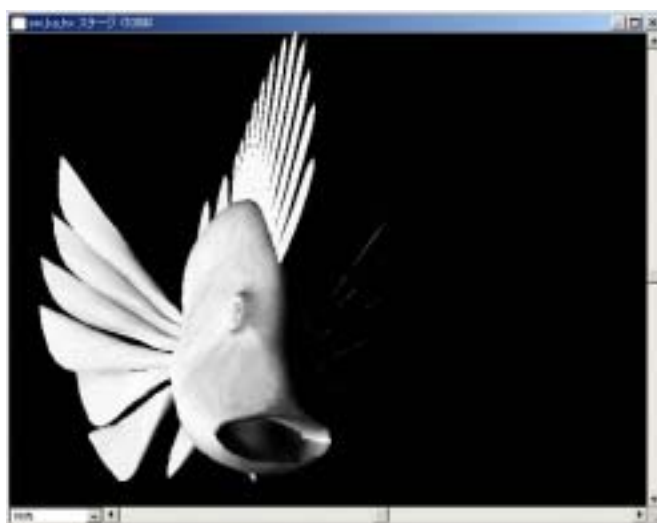


図 12 ShockWave3D の初期状態

モデルを回転する手法として、オブジェクト中心にカメラが移動するのではなく、モデル自体が回転する方法で解決できそうだが、全モデルの回転基点が一致していることが必要である。この作業にはカメラを回転するだけの作業より手間がかかり、回転方法としては効率が悪い。3D 空間上のカメラを任意に動かせる場合、そのカメラがモデルの裏側に行き着いたときには真っ暗なモデルを目にすることになる。3D 空間の表現を得意としている ShockWave3D においてはモデルが動くことで正面の視点から影側が見えなくなるのを回避するというはあまり意味を成さない。見せ方などで誤魔化せずに、モデルを十分に照らせる照明を追加しなければならない。その照明に関して、3DstudioMax にあるオムニライトは ShockWave3D では扱うことができない。そのため、オムニライトよりも情報量の多い指向性ライトでモデル全体を照らすように配置することになる。以上がモデル単体を照明する場合についての手段だが、2.4 節にて評価したとおり、ShockWave3D がプログラムや 3D 空間を表現することに向いていることを考慮すると、3D モデルに照明を当てて不可視部分をなくす方法を探るよりは、3D 空間上に太陽や蛍光灯のような、空間全体を照らす照明をつけてしまったほうが得策と言える。照明自体が ShockWave3D シーンに必要な要素であることは挙げられたが、ShockWave3D ではポリゴン数が少なくても、照明による最適化を要求されるほど粗い画像で表現されるわけではない。また、照明に工夫を凝らしてもモデルが綺麗に見えることはなく、単純にモデルの明度が変わるだけだった。これはモデルがある程度の滑らかさを持っていることと、ShockWave3D が金属などの光沢表現

をしないため、他の Web3D 技術よりも照明の影響を受けにくいからだと考えられる。

### 3.2.2 シェーディング

3.2.1 節では照明を追加して表面の質感を補う手法を試したが、本項目ではシェーディングの方法を変えること質感の向上を図る。より高度なシェーディング手法を適用することにより、照明による最適化よりもはるかに大きな最適化の効果が現れることがと期待できる。ただし、シェーディング機能は Web3D 技術それぞれが持っているもので、モデリング中の操作などで作り上げるものではない。そこで、各 Web3D 技術がどのようなシェーディングが可能なのかを調査した。VRML については、フラットシェーディングでシェーディングされ、それ以外のシェーディング方法の指定はできない。VRML で描画されたモデルを図 13 に示す。無骨な感じの画像で、一昔前の 3D グラフィックスを見ているような印象を受ける。照明から照射された光はモデルの形状通りに反射される。テクスチャを用いた場合を除いては、3.2.1 節で挙げたような照明の追加をしない限り制作されたモデルがそれ以上のものに見えるということはない。

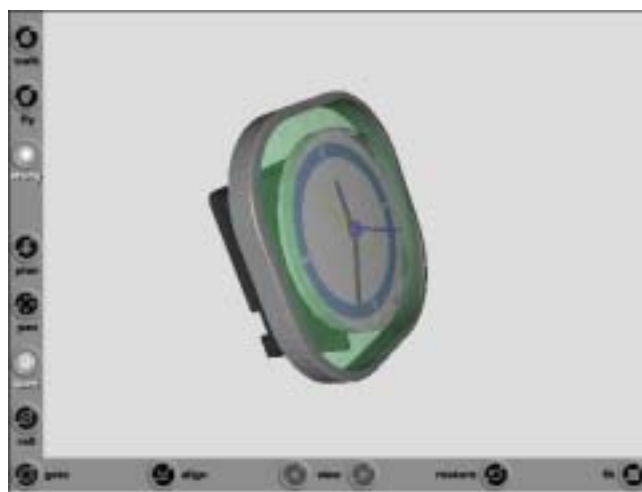


図 13 VRML

シェーディングの選択の余地がない VRML ではシェーディングの変更による最適化を図ることはできない。モデリングツールではプレビュー画面にスムーズシェーディングを用いてモデルを表現するものが多い。このようなツールを用いてモデルを制作する場合は、

モデリングツールのプレビュー画面をフラットシェーディングによるものに変更するか、それが不可能な場合はクリエイターが注意を払ってモデリングする必要がある。スムーズシェーディングでモデリングし、後からフラットシェーディングによってレンダリングされたモデルを見ると、モデルが予想外に粗悪な仕上がりになる場合がある。

Viewpoint では、7つのシェーディング方法を選択できる。そのうち“ Point ”と“ Wire ”については、表面のポリゴンを描かず頂点や線だけのモデルにする。“ Flat ”ではフラットシェーディングとほぼ同質だが、光沢や陰影を一切表現しないでシェーディングする。“ Texture ”では“ Flat ”に加えてテクスチャの描画と、極僅かな光沢が描画される。“ Lightmap\_Texture ”がデフォルトのシェーディングと同じもので、“ Lightmap ”と“ Lightmap\_TexMod ”もこれに近い画質になる。いずれも、“ Flat ”シェーディングに改めて照明を当てて光を再計算して光沢を表現したような質感で、一般的な 3D グラフィックスと比較して、イラストやセルアニメに近い画質を実現している。“ Lightmap ”と“ Lightmap\_TexMod ”について、“ Lightmap ”では、テクスチャが描画されない、“ Lightmap\_TexMod ”ではテクスチャの下のマテリアルの色が透けるといったわずかな違いがある。それぞれのシェーディングを施したモデルを図 14、～図 20 に示す。Viewpoint ではポリゴンの粗い粗悪なモデルでも、デフォルトのシェーディングで充分綺麗な画像になるので、画質を向上させるためにシェーディングの手法を選択することはないと考えられるがモデルの見せ方を選択できるので、これら 7つのシェーディングからの選択で Web コンテンツの目的に即したモデルを作ることができる。

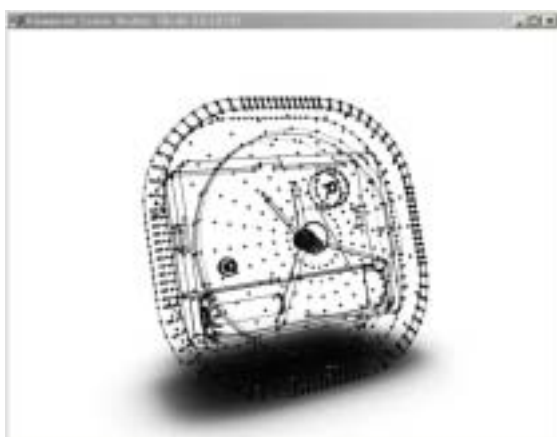


図 14 Point



図 15 Wire



图 16 Flat



图 17 Texture



图 18 Lightmap\_Texture



图 19 Lightmap



图 20 Lightmap\_TexMod

Cult3D は 4 つのシェーディング方法があり、“ Constant ”、“ Flat ”、“ Ground ”、“ Phong ” の 4 つからの選択が可能である。それぞれのシェーディングを適用したモデルを図 21 ~ 図 24 に示す。“ Constant ” は、Viewpoint での “ Flat ” 同様のシェーディングで、光沢を表現しないシェーディングである。“ Flat ” は VRML のシェーディングと同じ、3D グラフィックス全般でフラットシェーディングと呼ばれるものである。“ Ground ” では、隣接しているメッシュの位置や色の関係を含めて光の計算がされる、スムーズシェーディングが適用される “ Constant ”、“ Flat ” と比較すると格段に画質のなめらかさが向上する。ポリゴンの粗いモデルでもエッジが目立つということはほとんどない。“ Phong ” に関してはこの処理をメッシュ単位ではなくピクセル単位で計算する、鏡面反射シェーディングの基礎となっているシェーディング方法である。

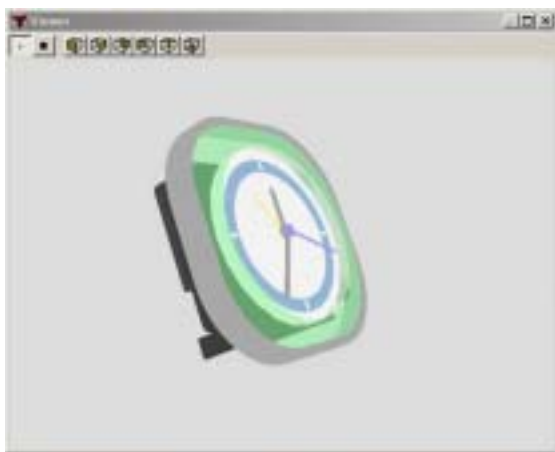


図 21 Constant

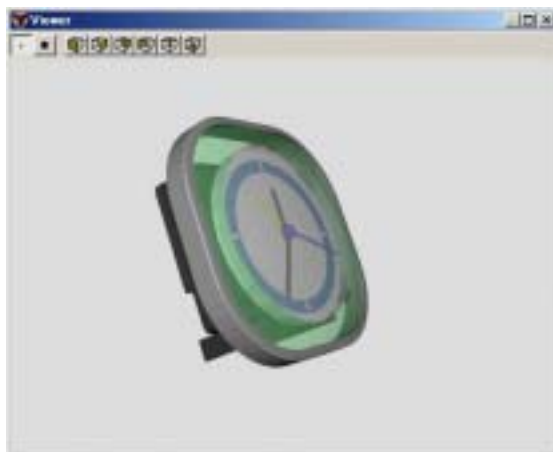


図 22 Flat

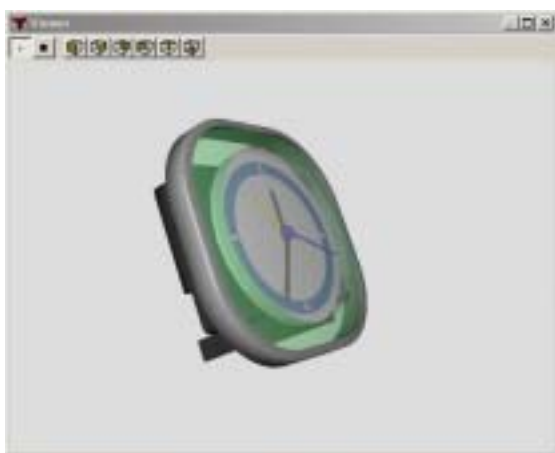


図 23 Ground

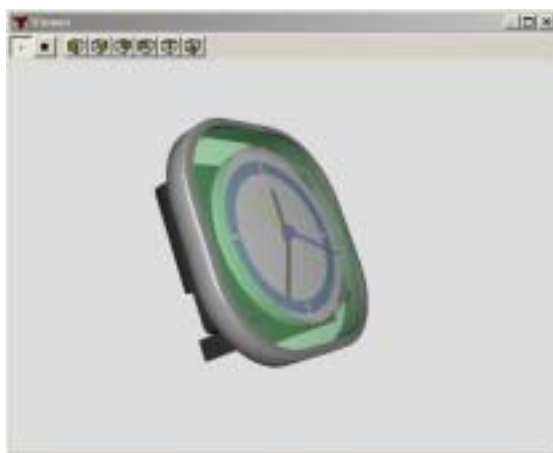


図 24 Phong

ShockWave3D には “ standard ”、“ painter ”、“ newsprite ”、“ engraver ” などがあり、これらは光の反射についてというよりも物体の見せ方についての表現となる。取り上げた 4 種類のシェーディングについて、図 25～図 28 に示す。“ standard ” は名の通りあまり特徴がない。綺麗とも無骨とも取れない無難なスムーズシェーディングによる画像になる。ShockWave3D の特徴となっているのだが、このシェーディング方法でも物体の光沢については他の技術ほど露骨には表現されない。“ painter ” では影が塗りつぶされ、絵画のような画像になる。“ newsprite ” ではテレビの空きチャンネルのような影で陰影が描かれ、“ engraver ” では粗い斜線で陰影が表現される。いずれの手法を用いてもそれによってエッジが目立つということはない。作られた 3D コンテンツに視覚的な効果を与えて、コンテンツに合った雰囲気を作れるシェーディングである。

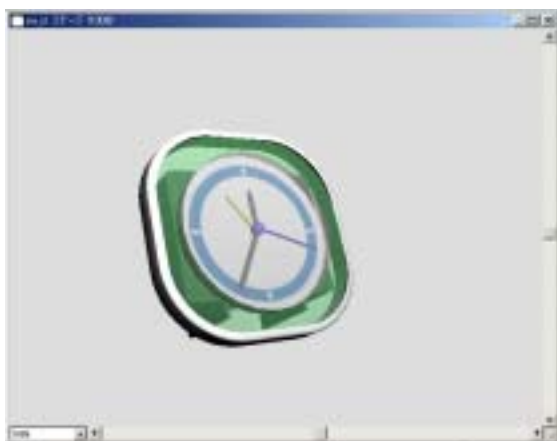


図 25 standard



図 26 painter

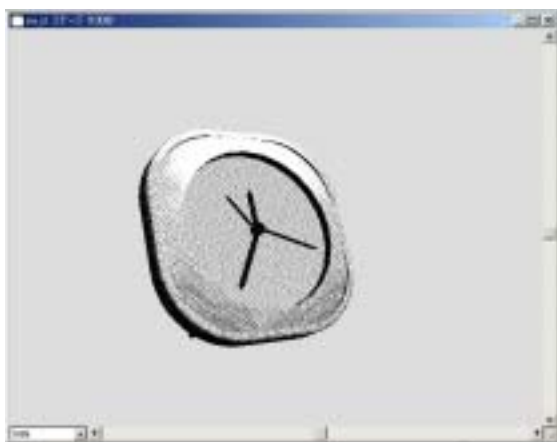


図 27 newsprite

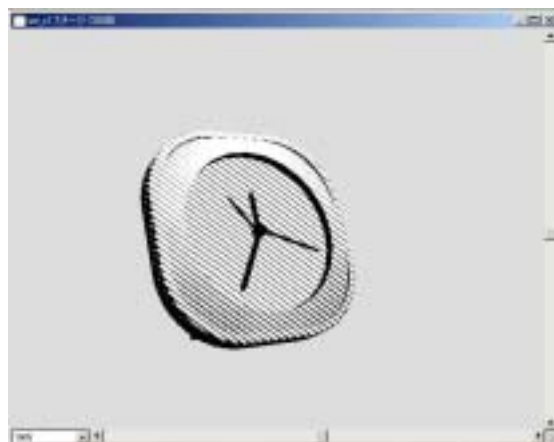


図 28 engraver



VRML では選択の余地のないシェーディングだが、Cult3D では大きな変化が見られた。シェーディングの方法によっては、物体の質感そのものが変わることがある。Viewpoint・ShockWave3D では物体の見せ方によってシェーディング方法を選択し、よりわかりやすく、または面白くモデルを表現できる。特に ShockWave3D 物体の質感よりも画質そのものが変化し、モデルの表現手法の効果を楽しむことができる。

### 3.2.3 マッピング

精密に作られたモデルと、簡素に作ったモデルとの差を最も小さくできるのがマッピングによる効果である。簡素なモデルでも、テクスチャを貼ることでモデルの陰陽のコントラスト差を小さくし、画質を向上させることができる。大部分の Web3D 技術ではマテリアル、透明度の表現と、ごく一般的な画像、そしてバンプマップを扱える。一般的な 3D グラフィックスについてはこれ以外にもディスプレイメントマップなど、モデルの形状にまで影響を及ぼすようなマッピングの技術があるが、処理に時間がかかる、コンピュータへの負荷が大きいなどの理由から、このような高度な手法を扱える Web3D 技術はほとんどない。今回取り上げた 4 種類のマッピングは全ての Web3D 技術で扱うことができる。

マテリアルと透明度の表現については、モデルのリアリティを向上させる以外の効果はなく、この 2 つの違いは画像が透けて表現できるかできないかの違いしかない。両者とも色情報と反射率の情報を持っている。プラスチックやガラスなどの素材をモデリングするとき、あるいはモデルを用いて図解するような場合には透過処理が必要となる。テクスチャを用いる場合に比べ、単純な色情報しか持っていないマテリアルは非常に小さな情報量で表現される。ただし、フラットシェーディングによってシェーディングされた場合はそのモデル本来の形状が露骨に表現されることとなる。シェーディング方法を変えるなどの手法を併用してモデルをうまく見せる工夫が必要である。

マテリアルでは表現が難しい模様などについては JPEG などの一般的な画像をテクスチャとして用いる。画像を上から貼り付けることでその下にあるマテリアルの持つ特徴はなくなってしまう。マテリアルの特徴を生かしたまま画像を貼り付けたい場合は、背景に透明度を持つ画像を用意する、テクスチャを最小限の範囲にとどめるなどの工夫の必要がある。人は簡素なモデルと精密なモデルの差は、明るい面と暗い面との間がどれほど滑らか



に変化をしているかで見分けるが、テクスチャを適用すると面の明暗差が小さくなり、簡素なモデルでも精密なモデルとの見た目の差がほとんどなくなる。本来明暗差があるところのリアリティが失われてしまう場合は、テクスチャの画像のその箇所を暗くしてしまえばよい。また、色を暗くする程度では解決できない場合にはバンプマップを用いる方法がある。

バンプマップとは、テクスチャとして用いる画像に表面の反射率の情報を加えたもので画像の箇所によってその反射率を変化させることができる。その結果物体の表面に明るい部分、暗い部分が生まれる。多くの場合、暗い部分がくぼんでいるように見えるようになる。ただし、バンプマップはポリゴンの形状を変形させて凹凸を表現するわけではないので、カメラがモデルに著しく接近したときや、輪郭線がはっきり見えてしまう状態にある時、または極端に大きな突起を表現する場合に、大きな違和感を生んでしまう。基本的な形状さえモデリングされていれば、バンプマッピングは細かな凹凸を表現するには非常に効果的な手法で、軽い処理で表現できる。小さな凹凸ならばほとんどの場合に有効だが、特に携帯電話のボタンのくぼみなどどのような角度から見ても輪郭線が見えないものに対しては非常に有効である。

3D モデルにテクスチャを貼るということは、情報量が画像分だけ増えるということである。3D グラフィックスとしてのクオリティを保つためにはテクスチャにもそれなりの解像度の求めることになるので、闇雲にテクスチャに頼るのはファイルサイズの増大につながる。単色で表現できるところは素直にマテリアルを利用した方がよい。特にバンプマッピングに関して、本来テクスチャを貼る必要がない凹凸部分については、ポリゴンで凹凸をつけるのとバンプマップを使うのとどちらが効率的にモデルを再現できるかを考える必要がある。

これらのマッピングの効果は全ての技術で効果が上げられた。結果を図 29 に示す。これまでに有効な最適化手法が見当たらなかった VRML についても、簡素なモデルと精密なモデルの差がほとんどなくなって見える。むしろ、精密なモデルはポリゴン数が多く、多くの座標について計算するため処理項目が膨大になる。そのため、精密なモデルの方が、操作しにくいように感じられた。また、VRML は画像ファイルを改めて読み込み、モデルに合わせて画像の座標や大きさを計算することが必要になることも原因の一つだと考えられる。Cult3D についても簡素なモデルと精密なモデルの差が見られなくなった。Viewpoint と ShockWave3D についても完全に見た目の差はなくなったが、テクスチャがない状態で

も簡素なモデルと精密なモデルの見た目の差が小さかったので、テクスチャによる効果が VRML ほど大きいようには感じられない。

質感の補完には大きな効果をもたらすマッピングだが、VRML についてはファイルサイズに大きな問題が出た。ポリゴンの情報に加えて画像の座標に関する情報が増え、ファイルサイズがおおよそ 2 倍に増える。VRML ではこの他に、JPEG などのテクスチャの画像が用意される必要があり、テクスチャのないモデルとは全く比較にならない量のデータの受信が必要とされる。Viewpoint、Cult3D、ShockWave3D では各種ファイルの中にテクスチャ画像が圧縮保存され、ファイルサイズの増加は数十キロバイト程度であった。また、Viewpoint、Cult3D ではモデリングツールからの書き出し中に画像を圧縮するダイアログが表示され、画像の情報量を調節することができた。

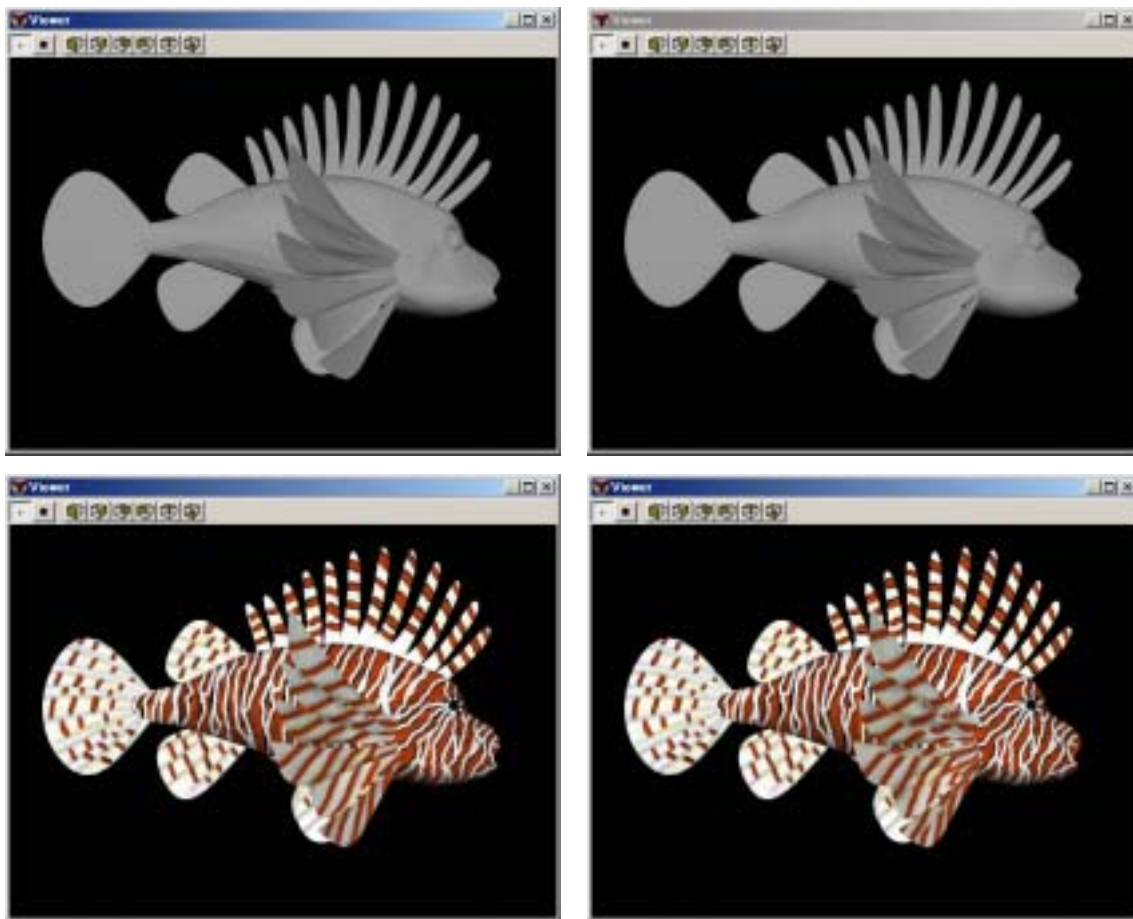


図 29 テクスチャによる最適化

最適化の評価をするために用意した Cult3D を用いたコンテンツで、魚にマッピングを施

し、更に Ground シェーディングで書き出した際、ノートパソコンではテクスチャ画像が表示されない場合があった。合計で 6 万ポリゴンに及ぶモデルに対してのシェーディングの処理による影響と考えられるが、マッピングを施す際に不要に高度なシェーディングを適用すると害が生じることがわかる。

### 3.3 機能

まず、最適化機能に関して、各技術が持っているものを 3.3.1 節に示す。これは各 Web3D 技術が独自に持っているもので比較できないと判断し、記述のみにとどめることにした。次に編集可能なスクリプトファイルについて、一度書いたスクリプトをインスタンス化して、同じ文を複数回の記述することを避けられるかなど、スクリプト部分についての機能を 3.3.2 節に示した。

#### 3.3.1 最適化機能

VRML は LOD 機能を備えている。これはあるモデルからカメラが離れたときに、そのモデルをあらかじめ用意された簡素なモデルと入れ替えて、処理を高速化するための機能である。VRML そのもののファイルが巨大で、そこにモデルを追加して読み込ませることになるので、有効な場合は少ないが、森林のように同じモデルを多数読み込む場合、この機能を用いて処理を軽くすることができる。手前の木、奥の木の 2 つを複数配置し、森を表現する。複数のファイルを読み込むことになるので、ダウンロード時間は増えるが、処理の軽い木を用意することで、Web3D コンテンツ上の合計ポリゴン数を減らし、処理にかかる時間が大きくなるのを抑える効果が出る。

Viewpoint は 3D 空間についての最適化機能は持っていないが、モデル単体に対しての最適化機能や表現機能を備えている。3D モデルをより効果的に Web コンテンツ内に収められるような機能が多く、コンテンツ全体として無駄な記述を省くことができる。具体的には Z-Buffer や Viewpoint 内にカメラに水平なテキストを表示させるなどの表現に関する機能を持っている。また、最適化機能のほとんどは機能付加の記述をしなくても適用されており、デフォルト状態でもモデルが自然に見えるようになっている。モデルの情報が格納

されている mts ファイルについては、Streaming tuning studio というアプリケーションで再編集でき、ポリゴン数が必要以上に多い場合はここでモデルのポリゴン数を少なくすることができる。

Cult3D は Cult3D Designer での編集中に Wavelet という圧縮アルゴリズムを使用してテクスチャを含めてモデルの情報を圧縮することができる。テクスチャを圧縮する際、Wavelet では圧縮率を指定することができ、圧縮後の画質はプレビュー画面で確認しながら作業することが可能である。Wavelet はモデル自体の圧縮と、画像の圧縮とで別に圧縮率を変更でき、モデルの圧縮の際にはアニメーションとモデル形状の 2 項目についてそれぞれを別に圧縮することができる。また、Wavelet による圧縮のほかに 3DsutudioMax や Maya から Cult3D 形式で書き出す際に、メニュー中の Polygon reduction の項目でポリゴン数を削減できる。これについては Join coincident vertices 項目と併用することで、ポリゴン削減によってモデルに隙間が生じるのを防ぐことができるようになっている。

ShockWave3D では VRML 同様に LOD 機能を備えている。この他に、距離クリッピング機能の hither、yon を調整することで最適化を実現できる。例えば、“camera.yon” の値を低くすることによって、遠距離にあるオブジェクトをレンダリングしないように設定できる。これらの機能にはモデル単体に対しての最適化の効果はないが、Web3D コンテンツ全体におけるポリゴン数を落とし、処理速度の低下を抑える効果を持っている。これら最適化のテクニックについては[15]に例が掲載されている。

### 3.3.2 スクリプト

内容がテキスト形式で記述される編集可能なファイルを公開する必要があるのは、VRML と Viewpoint である。これら全てについてスクリプトをインスタンス化して、複数回同じ記述をしなくて済むようになっている。VRML については作ったモデルにモデル名をつけることで、そのモデルはそれ以降モデル名で読み込むことができるようになる。また、基本的な直方体や球体、円錐などは基本プリミティブとして呼び出すことができるので、単純な形状に関してはモデリングツールを使わずに呼び出した方がファイルの増加はわずかで済む。“Cosmo player”、“cortona VRML Client”などの VRML ビューワーでは gzip 形式で圧縮されたファイルの読み込みが可能で、本来巨大な VRML のファイルを圧縮

して公開することが可能である。ただし、圧縮ファイルであることを宣言するために拡張子を\*.wrz へと変更する必要がある。ほとんどのユーザは“cortona VRML Client”を使って閲覧しているようだが、全ての VRML ビューワーが gzip 形式を読み込めるわけではないことを忘れてはならない。

Viewpoint の編集可能なファイル (MTX ファイル) ではモデルの位置、アニメーション情報と 3D 空間の設定についての記述がされている。基本プリミティブもこのファイルに記述することで生成することが可能である。また、照明についてもこのファイル内に手続き型ライトを生成できる。アニメーションの記述がされるため、長く複雑なアニメーションをする場合はファイルサイズが大きくなっていくが、スクリプトをインスタンス化することでその増加は抑えることができる。VRML 同様、Viewpoint の MTX ファイルも圧縮が可能で、こちらは Viewpoint Media Compressor を使うことによって圧縮される。生成される MTZ ファイルには、内部に複数の MTX ファイルを格納できる。MTZ ファイルは編集することができないが、同ソフトウェアを用いることで圧縮された MTX ファイルを再び解凍することができる。

ShockWave3D には Lingo によるスクリプトの記述が必要となる。このデータをそのまま公開することはないが、効率的な記述によって最終的に公開する ShockWave3D シーンのファイルサイズを小さくすることが可能である。ShockWave3D では Lingo によって書かれたスクリプトを ShockWave3D シーンに読み込ませていく。モデル情報、カメラ動作、インタラクションについて、別のスクリプトを読み込んでいくことができ、作ったスクリプトを保存しておけばそれ以降は読み込むだけで、後のシーンを作成する際の編集が簡単になる。他の技術と同様に標準プリミティブの呼び出しが可能で、単純な形状のモデルならばモデリングツールからの書き出しより少ない情報量でモデルが生成できる。シーン内に多くのインタラクティブ機能を取り込めばそれだけスクリプトは長くなりファイルサイズが巨大化していくが、モデルのパーツやモーションを別のファイルに保管し、必要なときにロードするように記述することで、必要のないときにはダウンロードさせないように命令できる。スクリプトは ShockWave3D シーン生成時に圧縮され、それ以降編集することはできなくなる。

## 3.4 環境

動作環境について、各種 Web3D 技術において、閲覧に推奨される環境を 3.4.1 節に示した。環境が整えられていれば問題はないが、ここに挙げられる環境が一般家庭のパソコンの性能に近い場合は、その Web3D 技術を用いるのは時期的にまだ早いということになる。また、最低動作環境と推奨動作環境に大きな差があると判断された Web3D 技術についてはその点についても考察を加えてある。また、ファイルサイズにおいて、どの程度のモデルで、どの程度のファイルサイズが必要とされるのか。目安程度に本論文で用いたモデルの各技術で書き出した場合のファイルサイズを 3.4.2 節に示す。

### 3.4.1 動作環境

VRML ビューワー (Cortona VRML Client) は Microsoft Windows 95 / 98 / Me / 2000 / XP あるいは Windows NT 4.0、Macintosh では OS X までの全バージョンのオペレーティング・システム (以下 OS) 上で動作する。また、閲覧のためには Web ブラウザが必要であり、Internet Explorer 4.0 以降のバージョン、あるいは Netscape 4.0 以降のバージョンが求められる。処理するための CPU は Pentium プロセッサ 90MHz 以上が必要である。また、最小限 16MB のメモリが必要であり、プログラムファイル保存のため、ハードディスクに 6MB の空き容量が必要である。ディスプレイは 800 × 600 以上の解像度を持つもので、16bit 以上の色数での表示を推奨している。

Viewpoint Media Player 動作環境については、Windows マシンの場合、OS は Windows 95, 98, 2000, ME, NT4.x, XP のいずれでも良い。ブラウザは Microsoft Internet Explorer 4 以上のブラウザ、または Netscape Navigator 4.07 以上「6 は未対応」とのことだが、現行の Netscape 7.01 では標準インストールを選択した場合 Viewpoint Media Player がインストールされる。CPU は Pentium 166 プロセッサ (Pentium II) 以上のものが必要とされる。メモリは最低で 32MB RAM が必要だが、64MB 以上を推奨している。ディスプレイについては、カラーディスプレイ (24bit 推奨) またはそれ以上のモニタが求められている。通信速度にも推奨環境があり、28.8Kbps モデム以上を求めている。インストール時ハードディスクに 5MB の空き容量が必要だが、Viewpoint Media Player はネットワークに繋がっている状態で新バージョンの Viewpoint Media Player がリリースされていると自動でアップデ

ートをする。当然ながらこの際には相応のハードディスク空き容量が求められるので、5MBの空き容量さえあれば問題なく動作できるわけではない。Macintosh では Mac OS 8.5 から 9.x、OS X はクラシックモードのブラウザのみ対応している。ブラウザは Netscape Navigator 4.7 以上、または Microsoft Internet Explorer 5.x 以上 どちらでも良い。CPU に PowerPC 604 プロセッサ以上を求め、メモリに 128 MB RAM 以上、ハードディスク容量に 5 MB 以上の空き容量が必要とされる。

Cult3D については最低動作環境と推奨動作環境の 2 項目が記されていた。最低動作環境について、CPU は Pentium 133MHz または 100%互換性のあるプロセッサ。メモリは 16MB、ハードディスクに 1MB 以上の空き容量を求めている。Cult3D は非常に多くの OS 上での動作が可能で、Windows 95/98、Windows NT 4.0/2000、Mac OS (PowerPC)、HP-UX 10.10 以降、Solaris 2.4(SPARC)以降、AIX 4 以降、Linux 2.0 以降、以上のように記述されていた。ブラウザは Netscape 4.5+、Internet Explorer 4、Internet Explorer 5、またはこれ以降のバージョンでの動作が可能である。推奨環境について、CPU は MMX 搭載の最高速 Pentium または 100%互換性のあるプロセッサ、メモリは 128MB 以上とされており、それ以外の項目については最低動作環境と相違なかった。

Macromedia Shockwave Player 日本語版の再生環境について、Windows 上では、CPU について Intel Pentium または完全互換プロセッサ (Pentium II 以上推奨)、メモリに 32 MB 以上の空きメモリ。ブラウザは Netscape 4.0 日本語版以降、または Microsoft Internet Explorer 4.0 日本語版以降が必要である。Macintosh の場合、CPU は Power Macintosh (G3 以上推奨)、32 MB 以上の空きメモリ、ブラウザは Netscape 4.0 日本語版以降、または Microsoft Internet Explorer 4.5 日本語版以降が必要である。

いずれの技術に関しても、最低動作環境については問題ないと思われる。これらの動作環境に満たないパソコンでは現行のあらゆるアプリケーションの多くが扱えないことになる。このことから、Web3D コンテンツの表示自体は非常に質素な環境で扱えると言える。ただし、ここに挙げられたのは最低動作環境であり、記述を大きく上回る環境を持つことが望ましい。球体を 1 つ表示する場合と、商品紹介としてモデルを表示する場合とでは、その処理にかかる時間や負担が大きく異なる。また、3D 空間を表現する場合にはさらに多くの処理が必要とされる。Web3D コンテンツ上のモデルのポリゴン数や付加されたインタラクションによって、閲覧上問題が生じる環境も異なるため、閲覧の推奨環境を具体的に提示することはできないが、一般的なアプリケーションが扱えない程度の能力の場合には、

Web3D コンテンツを閲覧する際にも能力不足に陥る可能性があるだろう。恐らくその環境は、クリエイターにとっても予期していないユーザ環境であると思われる。

### 3.4.2 ファイルサイズ

本論文中で扱った 3D モデルのファイルサイズを記述しておく。時計のモデルについては、最適化前後のモデルに加えて、針が動くアニメーションを追加したものを記す。アニメーションによるファイルサイズの増分を比較することで、複雑なアニメーションをつけた場合に、どの Web3D 技術が優れているのか判別できるようにするためである。魚のモデルについては、テクスチャがあるものとないものである。テクスチャがあるもののファイルサイズ増分からテクスチャを利用することがファイルサイズに大きく影響することがわかる。

ファイルサイズが非常に大きな VRML だが、圧縮することでファイルサイズを格段に小さくすることができる。3.3.2 節に述べたとおり、全てのビューワーで圧縮されたファイルを読み込めるわけではないが、大きな効果を持っている。ただし、VRML ではテクスチャ画像は別途用意する必要がある。モデル単体を表示する場合について最もファイルサイズを小さくまとめることができるのは Viewpoint で、次いで圧縮した VRML、Cult3D、ShockWave3D、標準の VRML の順にファイルサイズが大きくなっていく。本論文で用いた全モデルのファイルサイズを表 1 に示す。また、表中のファイルサイズの下にダイヤルアップ接続、ADSL 接続を用いた場合のダウンロード時間を記した。ただし、現実的な転送量として、ダイヤルアップ接続で 17Kbps、ADSL の場合で 960Kbps での受信を想定している。



表 1 各種 Web3D のファイルサイズ

VRML(\*.wrl・\*.wrz)はテクスチャの画像ファイルを含まない

形式 モデル	VRML *.wrl	VRML 圧縮 *.wrz	Viewpoint *.mts + *.MTX	Viewpoin 圧縮 *.mts + *.MTZ	Cult3D *.co	ShockWave3D 3D_progbar.dcr + *.dcr
時計 LP	168KB 1m18s / 2s	36.2KB 17s / 1s	30.2KB 14s / 1s	25.9KB 12s / 1s	63.2KB 29s / 1s	164KB 1m16s / 2s
時計	564KB 4m23s / 5s	121KB 56s / 2s	65.7KB 30s / 1s	61.4KB 28s / 1s	208KB 1m36s / 2s	383KB 2m58s / 4s
時計アニメ LP	172KB 1m20s / 2s	34.1KB 15s / 1s	32.3KB 15s / 1s	26.4KB 12s / 1s	79.9KB 37s / 1s	164KB 1m16s / 2s
時計アニメ	600KB 4m40s / 5s	126KB 58s / 2s	67.6KB 31s / 1s	61.7KB 29s / 1s	225KB 1m44s / 2s	388KB 3m1s / 4s
カサゴ LP Texture なし	500KB 3m53s / 5s	121KB 56s / 2s	37.4KB 17s / 1s	36.3KB 17s / 1s	314KB 2m25 / 3s	244KB 1m53s / 3s
カサゴ Texture なし	1.00MB 7m46s / 9s	228KB 1m45s / 2s	70.2KB 33s / 1s	69.5KB 32s / 1s	589KB 4m33s / 5s	428KB 3m19s / 4s
カサゴ LP	964KB 7m29s / 9s	222KB 1m43s / 2s	106KB 49s / 1s	105KB 48s / 1s	357KB 2m46s / 3s	524KB 4m4s / 5s
カサゴ	1.92MB 14m54s / 16s	425KB 3m18s / 4s	153KB 1m11s / 2s	152KB 1m11s / 2s	631KB 4m53 / 6s	704KB 5m28s / 6s

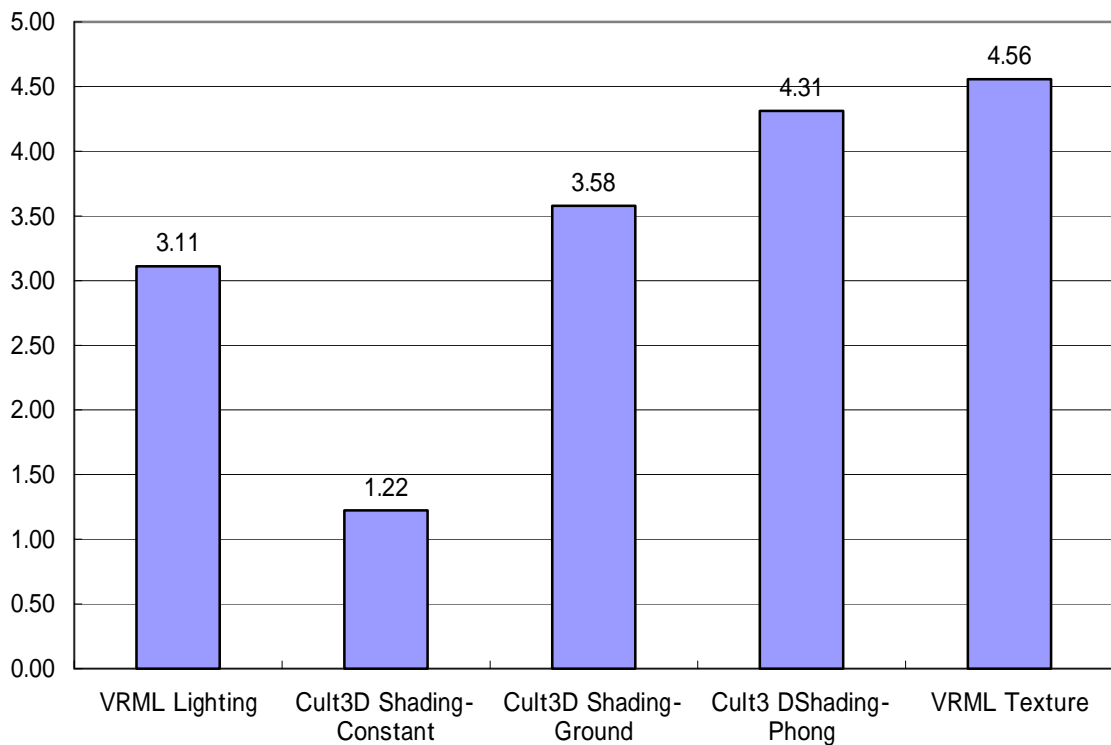
## 第4章 考察

第3章にて施した各種最適化についての考察を本項目でまとめる。描画項目に関しては5段階の評価アンケートを実施した。照明、Cult3Dのシェーディング、テクスチャについて、簡素なモデル、最適化手法を施したモデル、精細なモデルを見比べてもらい、簡素なモデルに手法をこらすことでどれほど精細なモデルに近い画質が実現できたか評価してもらった。評価の値の1は簡素なモデルよりも悪くなった、2は簡素なモデルと変わらない、3から5にかけて精細なモデルの画質に近くなったという評価をしている。アンケートの結果を表2に示す。

アンケートの結果、ライティングの評価は3.11で最適化としての効果が認められる最低値の3に非常に近く、最適化効果が弱いことが示された。ただし、照明の数をさらに増やし、各照明の輝度を変化させることで評価を高められることが考えられる。Cult3Dのシェーディングについては、“Flat”と比較して各種シェーディングがどれほど画質を高めたかを評価した。“Constant”を用いた場合の評価は1.22で、多くの被験者は画質が低下したと評価した。ただし、モデルの質感がセルアニメに近い場合、インタラク션을目的としたWebコンテンツなど、モデルのリアリティを求められないWeb3Dコンテンツでは有効な手法になりえる。“Ground”、“Phong”を用いた場合は、それぞれ3.58、4.31と高い評価となり、最適化の効果が認められた。“Ground”の評価については、“Phong”の画像を見たあとで一度つけた評価を下げ直すということが多く、被験者が“Phong”と比較したため評価が下がってしまっている。実質は4に非常に近い値で評価されていたと推測できる。テクスチャの評価結果は4.56で、ほとんどの被験者がテクスチャの最適化効果の高さを認めた。

表 2 アンケート結果

1 モデルの質が落ちた 2 最適化前と変化が感じられない 3~5 数値が高いほど最適化の効果が高い



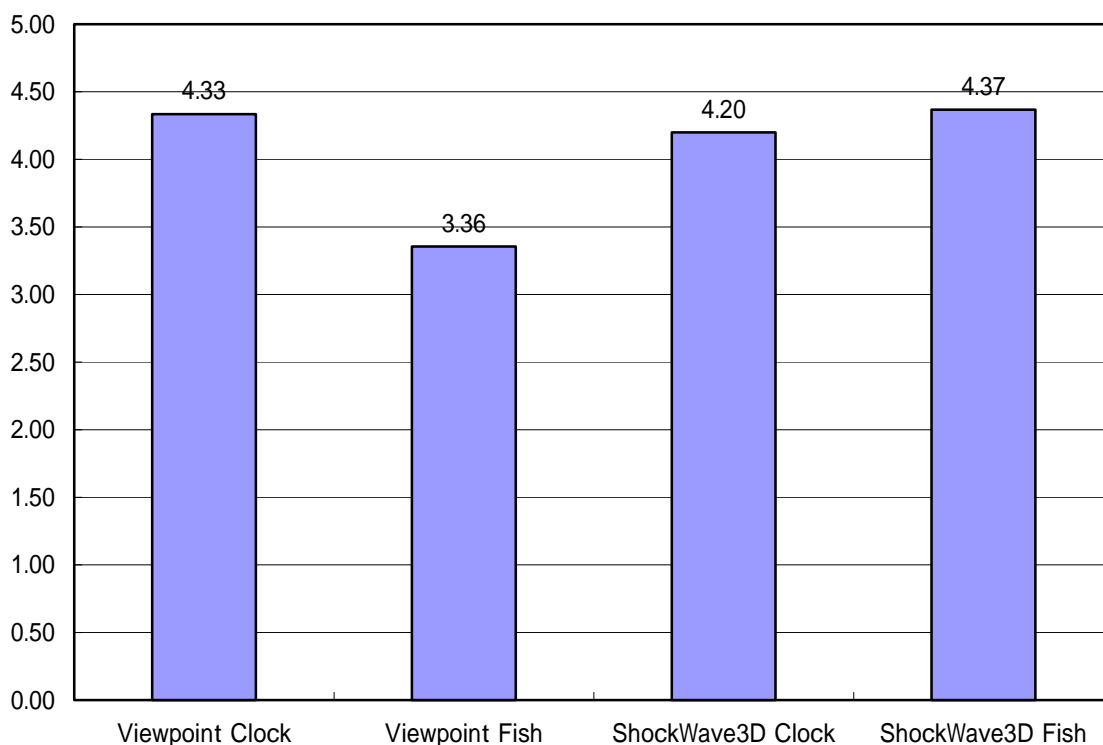
最適化の必要性が感じられなかった Viewpoint と ShockWave3D については、簡素なモデルと精細なモデルを見比べてもらい、簡素なモデルの画質を 5 段階で評価してもらった。1 は簡素なモデルの画質が明らかに劣っている、5 は精細なモデルとの差がわからない。以上の項目から Viewpoint と ShockWave3D に最適化の必要性がないことを客観的に評価した。そのアンケート結果を表 3 に示す

Viewpoint について、時計と魚それぞれの簡素なモデルと精細なモデルを見比べた結果は、時計が 4.33、魚が 3.36 であった。魚の評価が低いのが、ポリゴン削減時にヒレの形状が変化してしまったためと考えられる。輝度の高い照明の影響によって表面の形状が顕著に表れてしまうため、ポリゴン削減時にはこの点に注意を払う必要がある。ShockWave3D でも全く同じモデルを使って評価したが、その結果は時計が 4.2、魚が 4.37 と高い数値を示した。Viewpoint または ShockWave3D を用いる場合は、ポリゴンを削減しても画質に大きな影

響はなく、様々な最適化手法を模索しなくとも高品質な画像が提供できることが示された。

表 3 Viewpoint と ShockWave3D の簡素なモデルと精細なモデルの比較結果

1 精細なモデルと簡素なモデルとの質感の差が明らか ~ 5 両モデルの違いが判別できない



照明の効果による最適化は大きな効果を持つものではなかったが、VRML・Cult3D では表面を滑らかに見せることが可能であった。モデルに金属などの光沢を持つ部分がある場合には光沢が不自然になることが多く、照明に工夫を凝らす必要が生じるが、光沢の無いプラスチックなどをモデリングした場合には有効な手段である。ShockWave3D では照明を追加しないとモデルを見ることができないので、照明の追加という行為自体が必要である。Viewpoint については照明の追加が必要ない。表現方法の1つとして照明を追加することになるだろう。

シェーディングによる最適化は、Cult3D において最も大きな効果が期待できる。VRML ではシェーディングの選択の余地が無く、Viewpoint、ShockWave3D では表現手法の一環としてコンテンツの表現力を高めることができる。Viewpoint については、シェーディング

によって大きく質感が変化することはないが、シェーディング方法を変更することで、より目的に即した Web3D コンテンツが制作できる。いずれのシェーディング手法を選択した場合でも、シェーディングの処理は相当軽く、どのシェーディングを施しても処理速度に大きな差は見られない。従ってシェーディングの処理によってコンピュータへの負荷が大きくなることは考えなくても良いと言える。Cult3D のシェーディングは大きく質感を変化させ、“Ground”、“Phong”シェーディングを用いることで精細な画像を提供できる。ただし、画質を向上した“Ground”、“Phong”については“Constant”、“Flat”と比較すると極端に処理が重くなり、Web 上で閲覧することに問題が生じる可能性がある。“Ground”での処理は1つのモデルを表現する場合にはあまり問題ないが、広い3D空間やシミュレーションに用いた場合の動作は高速な三次元描画能力を持つグラフィックボードなしでは厳しい。マシン環境によって大きく左右されるので具体的な記述はできないが、おおよそ3万ポリゴンを超えるモデルや空間を現行のノートパソコンで扱うのは非常に難しい。“Phong”についても同様のことがいえるが、こちらは狭い3D空間でも処理が重く感じられる。ShockWave3Dについても、Viewpoint 同様に表現手法としてシェーディングを変更する機会が多い。ほとんどの場合、モデルの質感は損なわれることになるが、モデルの見え目が大きく変化することで簡素なモデルでも画像の粗さが目立たなくなる。

マッピングについては、すべての Web3D 技術において質感の向上を図ることができる。ただし、テクスチャ画像によってファイルサイズが増加することになる点のみ注意を払う必要がある。

最適化機能はそれぞれの Web3D 技術が独自に備えており、VRML や ShockWave3D は閲覧時の負荷を軽減する機能、Viewpoint は少ない情報量でモデルを精細に表現する機能を備えている。Cult3D は内部に多くの機能を備えており、最適化機能としてはファイルの圧縮機能がある。

スクリプトについては、VRML、Viewpoint、ShockWave3D において工夫が必要とされる。各技術ともスクリプトをインスタント化し複数回の長文の記述を避けることでファイルサイズの増加を抑えることができる。VRML のスクリプトファイルは gzip 形式で圧縮したものを読み込めるビューワー存在する。ファイルサイズを格段に小さくできるので非常に有効だが、gzip 形式を読めないビューワーを用いているユーザもいることを考慮する必要がある。

動作環境については、各 Web3D 技術とも非常に質素な環境でも動作する。ただし、3.4.1

節に挙げた環境はビューワーの動作環境なので、実際に Web3D コンテンツを閲覧する場合にはこれを大きく上回る環境が推奨される。

ファイルサイズについては、Viewpoint と圧縮した VRML のサイズを小さくまとめられる。ファイルサイズが大きくなる Clut3D と ShockWave3D だが、1 つのスクリプトや機能を複数のモデルに適用できる点や、別ファイルのモデルを読み込むことが可能な点から、Web3D シーンが複雑な 3D 空間など場合には、Viewpoint よりも小さなファイルサイズにまとめることができると考えられる。

## 第5章 結論

これまでの各 Web3D 技術の評価、最適化手法の適用の評価から、各種 Web3D 技術の差異が明らかになってきた。そこから、どのような場合にどのような技術を用いるべきなのかを述べる。Web クリエイターはパソコンの性能と通信速度に関して統一がとれないことを踏まえて、表現したい Web3D コンテンツに相応しい Web3D 技術を用いる必要があると言える。表現に相応しい Web3D 技術を知り、的確な Web3D 技術を用いてコンテンツを制作することにより、ユーザが Web コンテンツを閲覧する際その環境が好ましい状態でなかったとしても、画像がなかなか表示されない、動作が鈍いなどのストレスを小さくすることができる。本来、Web3D 技術は 2D コンテンツ以上のインタラクションを備えることができるという理念の下に作られており、的確な Web3D 技術を用いることは、コンテンツが持つ魅力をより大きく引き出す要素になると考えられる。

VRML では、多彩なカメラワークと LOD 機能を用いて、3D 空間を移動する場合の使い方が考えられる。ただし、標準の状態ではカメラの動作が非常に遅く VRML ビューワーのカメラに関する設定を変更することが望ましい。この点は閲覧者が決定するもので、クリエイターが設定を変えることはできないが、設定の変更を促しても良いだろう。もしくは、クリエイターが自らカメラの挙動を思惑通りに制御するアイコンを VRML 上に作るようになる。フラットシェーディング以外にシェーディング方法の選択の余地がないため、質感では他に劣るが、他のシェーディング方法よりも処理が軽いため、質感を無視できる状況ならば VRML も有効に使うことができるだろう。その際にも照明に工夫を凝らし、3D 空間やモデルが無骨にならないように注意を払うことが求められる。Web3D コンテンツを制作する際、スクリプトのインスタンス化や LOD 機能を用いて効率的に VRML を編集できるが、他の Web3D 技術を用いる場合よりもファイルサイズが大きいこと、テクスチャのファイル別途用意しなければならないことを考慮し、できる限り少ないモデルで 3D 空間を作るように心掛ける必要がある。モデル単体や、シミュレーションに用いる場合は質感の不足、表現力の不足などから、他の Web3D 技術を用いた方がずっと効率的に、美しく表現できると考えられる。

Viewpoint は、モデル単体を表現するときに非常に有効な技術である。質感の表現に優れ、他の Web3D 技術を用いた場合では露骨な 3D グラフィックスに見える画像でも、3D グラフィックスが Viewpoint によって柔らかかで好感の持てる画像になる。また、ポリゴン数を

大きく落としたモデルについても他の Web3D 技術のほど歪んだような画像にならず、滑らかに描かれる。ただし、ポリゴン数を落としすぎると輪郭が粗くなる場合があるので、曲面を輪郭に持ったモデルを作る場合には、この点には注意を払うべきである。また、ファイルの圧縮率が高く、生成される 2 種類の Viewpoint ファイルは合計しても他の Web3D 技術のファイルよりもずっと小さい。ファイルサイズが小さいということは、ユーザが閲覧する際のダウンロード時間が短いということで、ユーザのストレスになりにくいということに直結する。処理も非常に速く、アニメーションなどのインタラクションを HTML から制御させやすい点などを活用することでよりインタラクティブ性に富んだコンテンツを作ることができるだろう。しかし、3D 空間の表現やシミュレーションに用いる場合、モデル毎の明暗差に違和感が出る、MTX ファイルが肥大化するなどの問題が起こる。また、カメラの挙動に関して、設定によってウォークスルー型の移動をさせることもできるが、移動範囲や操作に限界があるなど、自由に操作できるとは言えない。このような 3D 空間上を移動するコンテンツでは、Cult3D や ShockWave3D といった他の技術を利用した方が良いだろう。

Cult3D は VRML を進化させたような技術で、物体の質感が似通っている。スムーズシェーディングが可能な点で、VRML を大きく上回るリアルな表現が可能である。表面が金属の商品紹介については Viewpoint に匹敵する数のコンテンツで用いられている。パーティクルアニメーションを扱えるため、普通のポリゴンモデルでは表現できない自然現象などを 3D グラフィックスで表現できる。他の技術にない最も大きな特徴は、モデルを形成しているメッシュの形状を変形させる、パーテックスアニメーションに対応している点で、生物の関節の動きなど、滑らかさを求められるアニメーションを綺麗に表現できる。パーテックスアニメーションを用いずに関節を動かすと、関節の内側のメッシュ同士がめり込みあってしまうため幼稚なアニメーションになってしまう。以上の点から、Web3D コンテンツをシミュレーションの表現として用いる場合には Cult3D を用いることで高度な表現が可能となる。非常に高機能な Web3D 技術だといえるが、フォンシェーディングを多くのモデルに用いる、あるいはシミュレーションとして実際に用いるとなると、グラフィック性能に改造を加えない限り、現在の市場品のパソコンでは少々無理がある。今後のパソコンの能力の向上と、通信環境の整備がされた場合に、汎用性が高く、多くの場面で用いられる Web3D 技術だと考えられる。

ShockWave3D は他の技術とは一線を画しており、モデル表現よりも Web コンテンツの



インタラクションの一環として用いることが想定されている。シェーディングについても、物をよりよく見せるためではなく、物をより面白く見せるための技術として扱われている。ShockWave3D シーン制作に必要な言語である Lingo がプログラムとして機能し、クリエイターは Lingo のプログラムによって、Web3D コンテンツをゲームやシミュレーション用に作成することが可能である。他の Web3D 技術で別途ダウンロードしなくてはならないプログラムファイルを ShockWave3D の一部として圧縮保存できるので、そのファイルは他の技術よりもずっと小さくなると考えられる。パーティクルアニメーションや LOD 機能などの最適化機能を持っているが、これらはシミュレーションとしての利用の他に、ShockWave3D を Web 上のアプリケーションとして扱った場合にも表現や負荷の減少に大きな効果を得ることができると考えられる。

以上のような Web3D 技術の特徴や機能を把握し、Web3D コンテンツの趣旨に合わせて使い分けるように心掛けることが、クリエイターにとっても、ユーザにとっても好ましい Web3D コンテンツを生み出すことになると考えられる。

現状での Web の利用目的は、コミュニティ、企業の商品紹介、インターネットショッピング、インタラクションがほとんどである。展望として、今後の Web3D コンテンツは以下 3 項目のように発展していくと考えられる。1 つ目はコミュニティ機能として、現在でもわずかにパソコン上でテレビ電話が使われているが、自分の映像ではなく、3D キャラクタを用いてコミュニケーションをとる手段も考えられる。子供たちには前者よりも受けが良いと考えられる。2 つ目はインタラクションの向上である。現在、Web3D コンテンツでは簡単な 3D モデルしか扱っていないが、いずれパソコンの性能と通信回線の整備が向上すれば、非常に高度なコンピュータグラフィックスを扱えるようになる。映画のような高品質の画像を Web 上のコンテンツとして扱えるようになるだろう。また、インタラクティブ性が増せば、現在あるオンラインゲームを Web ブラウザ上で表現することも可能になると考えられる。3 つ目に、インターネットショッピングに商品紹介が兼ねられた Web コンテンツが考えられる。現在のインターネットショッピングでは 2D 画像による商品紹介がほとんどだが、Web3D 技術を用いてより実物に忠実な画像を表示することにより、利用者の購買意欲が増すと考えられる。現在の Web3D コンテンツでは電化製品を扱う場合がほとんどだが、Web3D のインタラクティブ性を利用すれば、家具の販売において購入者の部屋に近い環境の部屋に家具のモデルを配置するショッピングサイトでは大きな効果を発揮すると考えられる。

## 謝辞

本論文執筆・研究にあたり、適切な助言と指導をして頂いた東京工科大学の渡辺大地氏、金尚泰氏、和田篤氏、西井美甫氏、また 3D モデルを提供して頂いた東京工科大学 AQUA-PROJECT の諸氏に厚く感謝の意を表す。

## 参考文献

- [1] 南 佳孝, VRML の達人 Web 工房シリーズ, 森北出版, 1999
- [2] VRML Worlds, <http://www.atom.co.jp/vrml2/index-j.html>
- [3] 荒屋研究室福岡工業大学 VRML VRML97,  
<http://www.fit.ac.jp/~araya/jis/index.html>
- [4] Viewpoint MTX/XML オーサリングガイド, <http://www.viewpoint.com/index.html>
- [5] Viewpoint XML Reference Guide, <http://xml.viewpoint.com/index.html>
- [6] Stream Tuning Studio ユーザーガイド,  
[http://richmedia.kgt.co.jp/dev/doc/japandoc/utility\\_pdf/streamtuningstudio.pdf](http://richmedia.kgt.co.jp/dev/doc/japandoc/utility_pdf/streamtuningstudio.pdf)
- [7] ふかの あきお, WEB3D 自由自在, ラピュータ, 2001
- [8] メロンメディアラボ, Viewpoint で作ろう!Web3D, アゴスト, 2001
- [9] サプリン, 佐野昌己, Director8.5 パワー・クリエイターズ・ガイド パワー・クリエイターズ・ガイド, アスペクト, 2001
- [10] 大重 美幸, DIRECTOR 8.5 Shockwave 3D オーサリングガイド, オーム社, 2000
- [11] 大重 美幸, Lingo スーパーマニュアル DIRECTOR 8 対応, オーム社, 2000
- [12] Isaac Victor Kerlow, 渡部晃久, コンプリート 3D グラフィックス, エムディエヌコーポレーション, 2001
- [13] Simon Danaher , 3D グラフィックスデザイン Design=directories, グラフィック社, 2002
- [14] Michael O ' Rourke, 袋谷 賢吉, 大久保 篤志, 3 次元コンピュータ・アニメーションの原理-3D コンピュータグラフィックスによるモデリング、レンダリング、アニメーション, 近代科学社, 2000
- [15] Macromedia, 3D コンテンツのファイルサイズ、パフォーマンスとストリーミングの最適化,  
<http://www.macromedia.com/jp/support/director/ts/documents/dr0289.html>