2007年度　卒　業　論　文

3DCGにおけるキャラクターの表示サイズによる漫画的簡略化表現手法

指導教員：渡辺 大地講師

メディア学部 ゲームサイエンス
学籍番号　M0104282
地神 知哉

2007 年度　　卒　業　論　文　概　要

論文題目

3DCG におけるキャラクターの
表示サイズによる漫画的簡略化表現手法

メディア学部 **学籍番号：M0104282**	**氏名**	地神 知哉	**指導教員**	渡辺 大地講師

キーワード	3DCG、LOD、MIPMAP、アニメ

近年、手描き調の画像を 3DCG を用いて再現するノンフォトリアリスティックな表現の研究が盛んに行われている。それに伴い 3DCG を用いて様々な漫画や 2D アニメーション特有の表現を再現する研究が行われてきたが、漫画や 2D アニメーションの表現を取り入れることによって生じる問題がいくつかある。その問題のうちの 1 つとして、画面上に小さく描いたキャラクターの表現がある。3DCG アニメーションにおいてキャラクターを画面上に小さく描いた際に、形状が細かい部分の輪郭線が重なり黒く潰れてしまう。そのためキャラクターの特徴がわかりにくくなってしまう。これは漫画や 2D アニメーションの特徴である常に太さが一定な輪郭線を描画したことによって生じた問題である。3DCG において 3D モデルを小さく描いた際の既存の技術として LOD(Level Of Detail) という技術がある。LOD は 3D モデルを大きく描いた際と小さく描いた際に描画するモデル切り替える技術である。しかし、LOD は描画処理を軽減するための技術であるため、3D モデルの切り替え前と後で見た目に変化を与えるものではない。そのため形状が細かい部分が潰れてしまうという問題は解決されない。それに対し漫画や 2D アニメーションにおいては、小さく描くキャラクターの形状を簡略化して表現する。簡略化することにより形状の細かい部分が無くなり、輪郭が潰れることがなくキャラクターの特徴が明確となる。そこで LOD の技術にこのような漫画的な簡略化表現を 3DCG アニメーションに取り入れることによって、3DCG アニメーションにおける問題点を解決する手法を提案し、その効果を検証した。

目 次

第 1 章

はじめに

1.1 研究背景と目的

近年、絵画調やセルアニメ調などの手描き調の画像を、3DCG を用いて再現するノンフォトリアリスティックな表現の研究が盛んに行われている [1][2][3][4][5]。それに伴い 3DCG を用いたアニメーションやテレビゲームにおいて、漫画や 2D アニメーション特有の表現を取り入れた作品 [6][7] が多く見られるようになった。既に 3DCG を用いて様々な漫画や 2D アニメーション特有の表現を再現する研究 [8] が行われてきたが、漫画や 2D アニメーションの表現を取り入れることによって生じる問題があり、その１つとして画面上に小さく描いたキャラクターの表現がある。それは、画面の遠方に小さく描いたキャラクターは小さく表示したときに、細かい部分の輪郭線が重なって黒く潰れてしまうという問題である。3D モデルの例を 1.1 に示す。

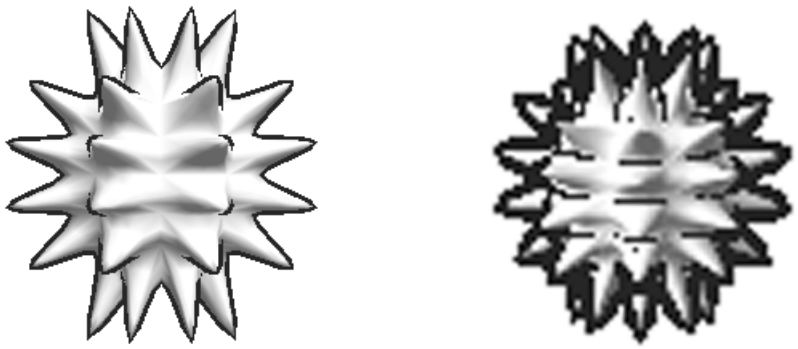


図 1.1: 3DCG の例

図 1.1 の右の 3D モデルは、左の 3D モデルを画面の遠方に小さく描いたものを拡大したものである。外側の細かい部分の輪郭線が黒く潰れてしまっていることがわかる。3DCG を用いたキャラクターにおいても同じように髪の毛などの形状の細かい部分が潰れてしまう。これは漫画や 2D アニメーションのように輪郭線の太さを一定にしているために、細かい輪郭を描くことができなかったことが原因である。これに対し、画面上で小さく表示したときにキャラクターの特徴をわかりやすくするために、始めから単純で特徴のわかりやすいデザインでモデルを用いる場合もある。その場合は画面上で大きく表示する際にディテールの低さが目立つなどの問題点がある。

既存の 3DCG の技術において、画面上に小さく表示したときの問題を解決するために利用できる技術として、以下のようなものが挙げられる。

- LOD（Level Of Detail）[9]
- MIPMAP[10]

LOD とはキャラクターなどのモデルを表示サイズによってディテールの違うものに切り替えていく技術である。大きく表示するときは高いディテールのモデル、小さく表示するときは低いディテールのモデルを表示する。

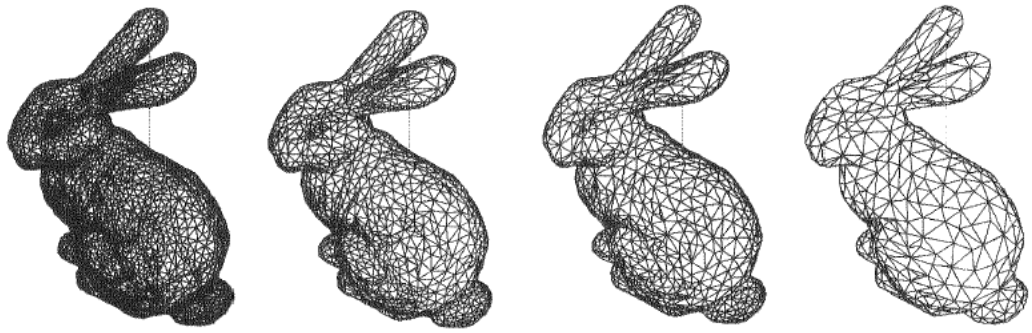


図 1.2: LOD

図 1.2 は、吉田らによる“ ポリゴンモデルを対象とする視界に依存する適応的表示 ”[11] から引用した LOD を説明した図である。左へ行くほど大きく表示する高いディテールのモデルであり、右へ行くほどは小さく表示する低いディテールのモデルである。このように低いディテールのモデルはポリゴン数が少なくなっている。表示サイズが小さくなり細かいポリゴンが表示できなくなったときに、低いディテールのポリゴン数の少ないモデルに切り替える。しかし、LOD は主にポリゴン数を減らして描画処理を軽減することが目的である。そのため縮小時に輪郭線が黒く潰れてしまう細かい部分を、潰れないように形状を考慮するものではない。MIPMAP は、画面上で遠方に小さく縮小されたテクスチャの色をぼやけさせる技術である。テクスチャをそのまま縮小するとモアレが発生してしまうので、それを抑制することが目的となる。図 1.3(a)〜(c) に例を示す。

(a) 元となる画像　　(b) 通常の縮小画像　　(c)MIPMAP 適用画像

図 1.3: MIPMAP

図 1.3(a) は元となる画像で、それを特別な処理をせずに縮小したものが図 1.3(b)、MIPMAP を適用したものが図 1.3(c) である。図 1.3(b) のように単純に縮小した場合は線が鮮明でなくなり形状が認識しづらいのに対して、図 1.3(c) は線をぼやけさせることによって大まかな形状が認識できる。しかし画像をぼやけさせるため輪郭が鮮明でなくなることや、ぼやけさせた際に中間色が生成することなどのため、漫画や 2D アニメーションのように輪郭線と色の境界を鮮明にする表現とは異なったものとなる。また、MIPMAP はテクスチャに対して行う技術なので、顔のパーツなどには用いることができるが、モデルの輪郭など、モデルの形状そのものにかかわる部分には利用できない。

このように、3DCG を用いた作品では小さく描いたキャラクターの輪郭を鮮明に描くことができない。それに対して、漫画や 2 D アニメーションにおいて遠方に小さく描くキャラクターは、手前に大きく描くキャラクターと比較し簡略化して描くことが多い。その簡略化例を図 1.4 に示す。

図 1.4: 漫画での簡略化例

図 1.4 は左のキャラクターは手前に大きく描いたもので、右のキャラクターは遠方に小さく描いたキャラクターを拡大したものである。右の小さく描いたキャラクターは左の大きく描いたキャラクターと比較し、以下のような特徴が挙げられる。

- 髪を大きな束として描いている。
- 顔のパーツを簡略化している。

大きく描いたキャラクターの髪は細い束で数も多いのに対し、小さく描いたキャラクターの髪は太い束で表現している。小さく描いたキャラクターは、顔のパーツの輪郭線の省略や簡略化などの特徴がある。このように、漫画や 2 Dのアニメーションにおいて小さく描いたキャラクターは単純に縮小したものではなく、キャラクターの特徴を残しながら簡略化して描く。小さく描くキャラクターを簡略化する理由として、以下のような原因が挙げられる。

- 輪郭線の太さがキャラの大きさにかかわらず一定である。
- 色がグラデーションせずに境界線が鮮明である。

まず、漫画や 2 Dアニメーションは手描きであるため、輪郭線がほぼ一定の太さで描く。そのためキャラクターを小さく描く際に、大きく描く際と同じように

細かい部分まで描き込もうとすると線が重なり、絵が潰れてしまう。また、2 Dのアニメーションにおいて色はグラデーションさせず境界線が鮮明であるため、小さくて見づらい部分の色をぼやけさせるなどの表現はできない。これらの理由から、遠方に小さく描くキャラクターも鮮明な輪郭線で描く必要があり、細かく描くこともできないので簡略化して描いている。簡略化して描くことによって、小さく描いたキャラクターも輪郭を鮮明にすることができる。

本研究では3DCGにおいて漫画や2Dアニメーション調の表現をしたときの3Dモデルの輪郭線に注目し、モデルを小さく表示したときに漫画や2Dアニメーションのように特徴をわかりやすく簡略化したモデルへと切り替える手法を提案する。まず初めに簡略化したモデルを用意しておき、視点からモデルまでの距離が一定より離れ小さくなったら用意しておいたモデルに切り替える。このときモデルを切り替える際に突然違うモデルに切り替えた場合違和感が生じる。そのため違うモデル同士を部分的に段階的に切り替えることで違和感を減少させる手法を提案する。最後に本研究の手法を適用したものと、従来の表現を比較し効果を検証する。

1.2 論文構成

本論文では、第2章では本研究で目標とする表現と、それを実現するための手法について具体的な手順などを示す。第3章では本手法を実際に適用し、検証を行う。第4章で本研究のまとめと今後の課題を述べる。

第 2 章

漫画や2Dアニメーションのような簡略化表現手法の提案

本章では、3DCG アニメーションにおけるモデルを縮小したときに輪郭線が重なって黒く潰れるという問題点を、漫画や 2DCG アニメーションのような簡略化表現を用いることによって解決する手法を述べる。2.1 節で本手法で目標とする簡略化表現、2.2 節で漫画や 2D アニメーションのような輪郭線の描画手法、2.3 節で本手法に必要な 3D モデルの特徴、2.4 節では 2.3 節で用意した 3D モデルを表示サイズによって切り替える手法について述べていく。

また、3D モデルの作成にはモデリングソフトの metasequoia[12] を使用し、プログラムの実装には OpenGL ベースの 3DCG クラスライブラリである FK System[13] を使用した。

2.1 目標とする簡略化表現

本手法では漫画や 2D アニメーションの特徴を考慮し、以下のような表現を目標とする。

- モデルの輪郭線の太さが表示サイズにかかわらず一定である。
- 小さく表示したときも細かい部分の輪郭線が黒く潰れない。

この２点を考慮しモデルの輪郭線に注目する。モデルを小さく表示したときにも輪郭線が細くならず一定の太さであり、かつ輪郭線が重なって黒く潰れないことを目標とする。そのためにモデルの形状を漫画や2Dアニメーションのように特徴がわかりやすいように簡略化する。

2.2 モデルの作成

ここでは、本手法で用いる3Dモデルについて説明する。まず一般的なLODには以下の2種類がある。

- モデルが小さく表示したときに用意しておいた低いディテールのモデルに切り替える手法
- モデルの表示サイズによって高いディテールのモデルのポリゴン数を動的に変化させる手法[14][15]

本手法では前者の手法を用いる。これは、漫画や2Dアニメーションで細い髪が束になって固まるなどのような簡略化した表現を、違う形状への変化とみなすためである。

初めに大きなサイズで表示するための高いディテールのモデルと、小さなサイズで表示するための低いディテールのモデルを用意する。このとき低いディテールのモデルは小さく表示しても輪郭が潰れてしまわないように、漫画や2Dアニメーションのように簡略化する。ディテールの高いモデルと低いモデルの形状は、切り替えても不自然でないようにある程度似せた形状になるように作成する。本論文では3段階のディテールのモデルを用意した。図2.1(a)は大きなサイズで表示する高いディテールのモデルであり、図2.1(b)、図2.1(c)はそれぞれ順にディテールを低くして、小さく表示しても輪郭が潰れないように簡略化したモデルである。3つのモデルは、それぞれある程度形状が重なるように制作する。図2.2は2種類のモデルを重ねて表示したもので、輪郭がある程度重なっていることがわかる。

(a) (b) (c)

図 2.1: 作成した 3 段階のディテールの 3D モデル

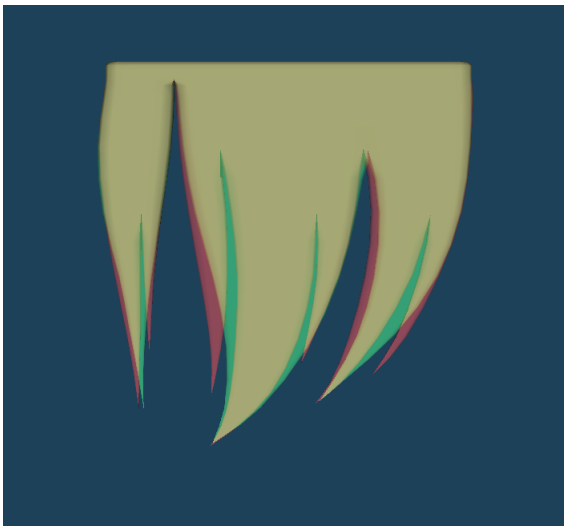


図 2.2: 違うディテールのモデルを合成した画像

2.3 輪郭線の描画手法

ここでは作成した 3D モデルに輪郭線を描画する手法を説明する。輪郭線を描画する方法として、鈴木による ”米国マンガ調セルシェーディングに関する研究 ” [16] において述べられている、3D モデルを引き伸ばし裏ポリゴンを利用する手法を用いた。以下にその手法の概略を述べる。

球体のモデルを用いて説明する。図 2.3(a) は球体モデルを正面から見た物である。これを視点上空から見下ろし、球を断面図にした物が図 2.3(b) である。図 2.3(b) ～(e) において、視点は図の下側にある物とし、実線部は視点から見て表示されて

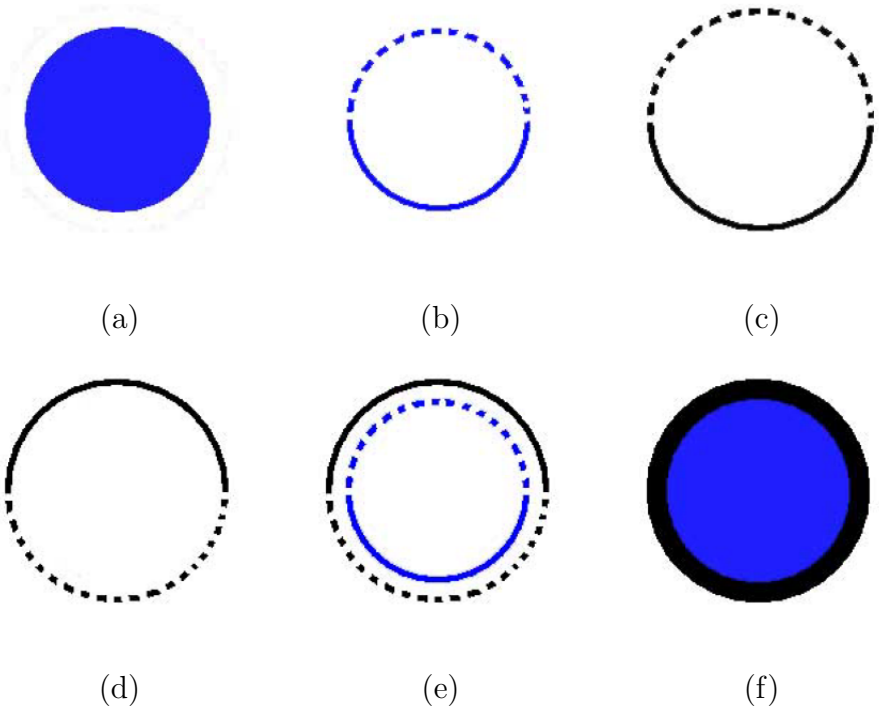


図 2.3: 輪郭線描画処理の過程

いる面、点線部は表示されない面を表す。輪郭線を描く処理は以下のように行う。

1. 元のモデルを頂点の法線方向に拡大し、面の色を黒くする。頂点の法線とはその頂点を構成要素に持つポリゴンの面の法線ベクトルの平均とする。

2. 拡大したモデルの面の表裏を反転させる。この処理によってモデルの視点から見て手前の面は表示されず、奥の面が表示される。

3. 拡大し、表裏を反転させたモデルに、元々のモデルを重ね合わせる。

図 (2.3c)〜(e) はこの処理を図示した物であり、図 2.3(c) は拡大し面を黒くした図、図 2.3(d) は面の表裏を反転させた図、図 2.3(e) は元々のモデルを重ね合わせた図となっている。この様な処理をした結果、処理後の図 2.3(e) のように、拡大したモデルは元モデルより大きくなってはみ出した部分だけが表示される事となる。図 2.3(f) が実際に視点から見た図であり、しっかりと球体モデルの輪郭に輪郭線を引いたように見える。

以上の処理を行った結果を以下の図 2.4 に示す。

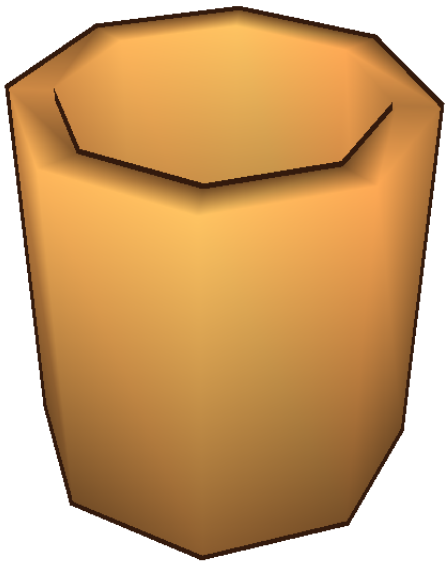
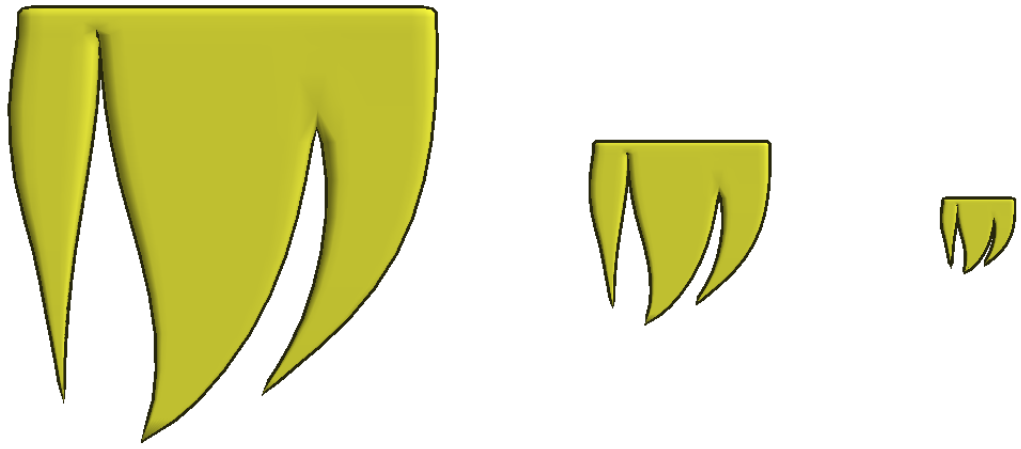


図 2.4: 裏ポリゴンを利用した輪郭線を描画した実行例

本手法では、モデルに常に一定の太さの輪郭線を描画する必要がある。上記の手法をそのまま適用するとモデルの拡大縮小に伴い、輪郭線も同時に拡大縮小してしまう。そのためモデルの表示サイズの拡大縮小に伴い、輪郭線用の裏ポリゴンモデルの元のモデルに対する拡大率を動的に変更する必要があるので、輪郭専用モデルの拡大をプログラム上で行う。輪郭線用モデルの拡大処理は以下の手順で行う。

- 元となるモデルの各頂点の法線ベクトルを得る。
- 輪郭線用モデルの各頂点の位置を法線方向に一定量移動する。このとき移動させる量は視点からモデルまでの距離に比例させる。

以上の処理を加えて、輪郭線をモデルのサイズにかかわらず一定の太さにしたものを図 2.5 に示す。



(a)　　(b)　　(c)

図 2.5: モデルの拡大縮小しても一定の太さの輪郭線

図 2.5(a)、(b)、(c) はそれぞれ同じモデルを拡大縮小したものである。このように、拡大縮小をしても輪郭線の太さは一定となっている。また裏ポリゴンを用いる本手法は、輪郭が元となるモデルの上に重なって表示されることが無いので、図 2.5(b) や (c) の毛の根元のようにモデル同士が近接している部分でも 2 本の輪郭線が重なって黒く潰れることはない。

2.4　モデルの切り替え手法

ここでは、用意しておいた複数のモデルを切り替える手法を説明する。本論文では 3 種類のディテールのモデルを使用しており、図 2.6 はその場合のモデルの切り替えの流れを説明した概略図である。

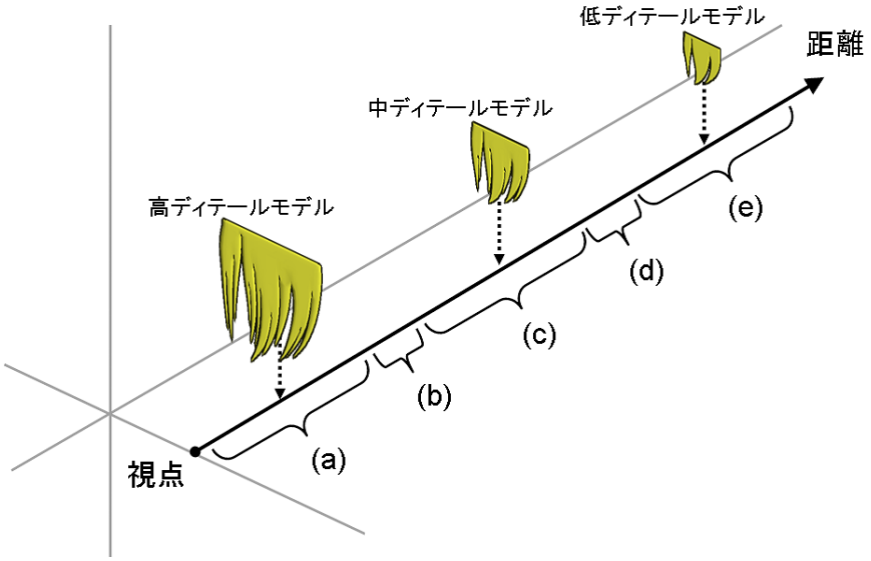


図 2.6: モデル切り替えの流れ

モデルの視点からの距離が図 2.6(a)(c)(e) の位置にあるときは、それぞれ用意しておいた表示サイズに合うディテールのモデルを表示する。ここではそれぞれのモデルを、高ディテールモデル、中ディテールモデル、低ディテールモデルとする。このとき、異なるディテールのモデルを単純に切り替えてしまうと、突然別の形状に変化したような違和感がある。それを回避するために図 2.6(b)(d) モデルを一気に切り替えるのではなく、徐々に間を繋ぐ手法をとる。この切り替えの間を繋ぐ手法については 2.4.1 節〜2.4.3 節で詳しく説明する。モデルを視点から遠ざける場合、まず図 2.6(a) では高ディテールモデルを表示しておき、図 2.6(b) の位置にあるときは高ディテールモデルと中ディテールの間を繋ぐ手法を適用する。図 2.6(c) の位置では中ディテールモデルへと完全に切り替える。図 2.6(d) の位置にあるときも図 2.6(b) 同様に、中ディテールモデルと低ディテールモデルの間を

繋ぐ手法を適用する。図 2.6(e) の位置では低ディテールモデルへと完全に切り替える。図 2.6(b)(d) における異なるモデル間の間を繋ぐ手法は以下の手順で行う。

1. モデルの輪郭線が重なり潰れやすい凹部分を検出する。

2. 形状を部分的に消去したモデルを生成する。

3. 視点からの距離によって、表示するモデルを切り替える。

2.4.1 節〜2.4.3 節でこの異なるモデル間を繋ぐ手法の手順 1〜3 について詳しく説明する。またプログラム実行時には、手順 1 と 2 は前処理として行っておく。

2.4.1 モデルの特徴点抽出

モデルの表示サイズを縮小したとき、初めに輪郭線が重なって黒く潰れる部分は輪郭が凹形になっている部分である。図 2.7 はその例を示したものであり、赤い丸で囲んである部分が輪郭が潰れやすい凹形の部分である。

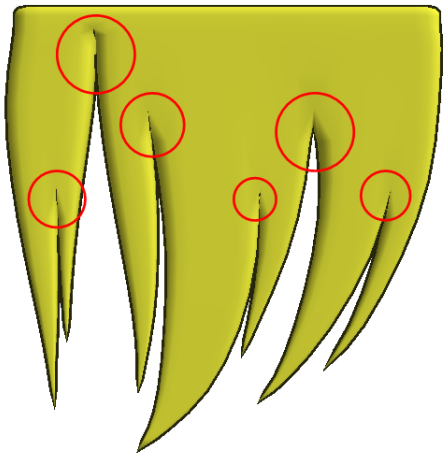


図 2.7: モデルの輪郭線が黒く潰れやすい部分

図 2.7 の赤い丸で囲んであるような凹部分を検出するために、各頂点の法線を利用する。頂点の法線ベクトルは、その頂点を構成要素に持つポリゴンの面の法線

ベクトルの平均とする。図 2.8 は接続している頂点とその法線ベクトルを表したものである。

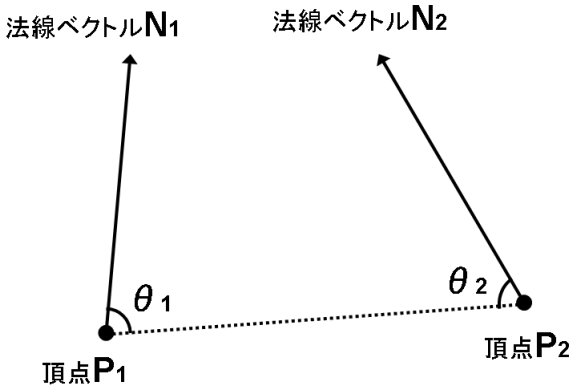


図 2.8: 接続した頂点とその法線ベクトル

モデル上のある頂点座標を $\mathbf{P}_1$、その頂点に隣接する頂点座標を $\mathbf{P}_2$ と置き、各頂点の法線ベクトルを $\mathbf{N}_1$、$\mathbf{N}_2$ と置き、$\overrightarrow{\mathbf{P}_1\mathbf{P}_2}$ と $\mathbf{N}_1$ のなす角度を θ_1、$\overrightarrow{\mathbf{P}_2\mathbf{P}_1}$ と $\mathbf{N}_2$ のなす角度を θ_2 と置く。このとき、θ_1 と θ_2 の合計が 180° 以下の場合に、$\mathbf{N}_1$ と $\mathbf{N}_2$ は互いに向かい合っている状態となり、凹形状の接続となる。また θ_1 と θ_2 の合計が小さいほど、より向かい合っている状態となる。全頂点について、各頂点の法線ベクトルとその頂点に隣接する頂点の法線ベクトルを比較し、θ_1 と θ_2 の合計が任意の角度以下となる隣接する頂点の数を求める。(2.1) 式はそれを数式化したものであり、A は 180° 以下の任意の角度とする。

$$A > \theta_1 + \theta_2 \tag{2.1}$$

各頂点について、(2.1) 式を満たす隣接する頂点が多い場合は、図 2.7 の赤い丸で囲んである部分のような輪郭が潰れやすい凹形の部分として、その頂点の情報を保存しておく。(2.1) 式の任意の角度 A を小さくすると、頂点同士の法線がより向かい合っている部分だけを検出することができる。

2.4.2 形状を部分的に消去したモデル生成

異なるディテールのモデルに切り替えるときに一気に切り替えずに異なるモデル間の間を繋ぐために、そのとき使用するモデルを新たに生成する。以下では高ディテールモデルと中ディテールを例に出して説明するが、中ディテールと低ディテールのモデルも同様の処理を行う。各モデルに対し、2.4.1 節で検出した頂点（以後特徴点と呼ぶ）を利用し以下のような処理をすることにより、形状を部分的に消去したモデルを生成する。

1. モデルの形状を部分的に消去するときの、消去する範囲の半径を求める。
2. 高ディテールモデルに対して、形状を一部消去したモデルを生成する。
3. 中ディテールモデルに対して、2 で消去した範囲以外を消去したモデルを生成する。

まず高ディテールモデルの全頂点についてそれぞれ最も近い特徴点までの距離を求め、その中で最も距離が長いものを求め、その距離を L とする。次に、異なるディテールのモデル間を繋ぐために新たに用意するモデルの任意の段階数 n を決める。本論文では $n = 6$ としているが、段階数 n が多いほど滑らかな異なるディテールのモデル間の切り替えとなり、少ないほど同時に全体が切り替わる。L を n で割った長さ l を求める。この長さ l が、高ディテールモデルと中ディテールモデルに対して形状を部分的に消去したモデルを生成するときの、任意の段階数 n の 1 段階目の各特徴点からの範囲の半径となる。このモデルの部分的な切り替えについての説明は 2.4.3 節で述べる。長さ l を求める数式を (2.2) 式に示す。

$$l = \frac{L}{n} \tag{2.2}$$

高ディテールモデルに対して、各特徴点から距離 l の範囲のポリゴンを消去したモデルを生成する。次に消去する範囲の半径を l の 2 倍、l の 3 倍としていき、l の n 倍まで同じようにポリゴンを消去したモデルを生成する。中ディテールモデル

は、高ディテールモデル側とは逆の部分を消去した、各特徴点から距離 l の範囲以外のポリゴンを消去したモデルを生成する。このように、形状の一部を消去した高ディテールモデルと中ディテールモデルを、それぞれ半径 l から l の n 倍までの n 個分生成する。図 2.9 は高ディテールモデルの形状を部分的に消去していく過程を図解したもの、図 2.10 は中ディテールモデルの形状を部分的に消去していく過程を図解したものである。

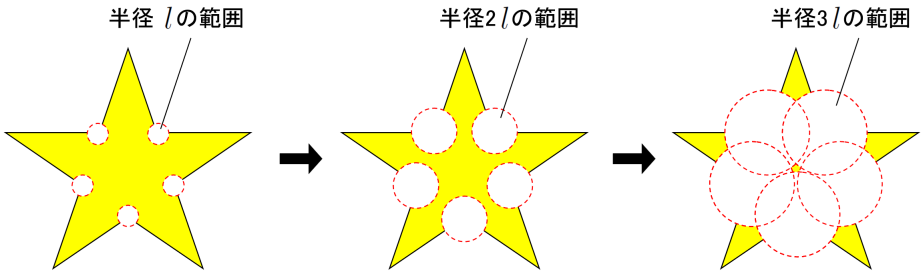


図 2.9: 形状を部分的に消去した高ディテールモデル

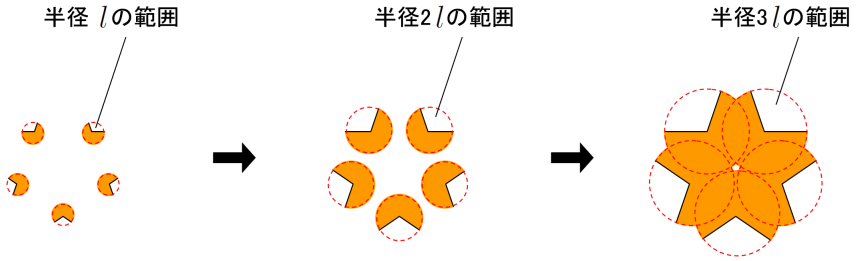


図 2.10: 形状を部分的に消去した中ディテールモデル

これらの処理を行ったモデルの実行例を図に示す。図 2.11 はそれぞれ高ディテールのモデルで、切り替え途中のモデルの一部が消去してある図である。図 2.12 も同様にそれぞれ中ディテールのモデルで、切り替え途中のモデルの一部が消去してある図である。

図 2.11: 形状を部分的に消去した高ディテールモデル

図 2.12: 形状を部分的に消去した中ディテールモデル

2.4.3　視点からの距離によるモデルの切り替え

高ディテールモデルから中ディテールモデルへと切り替えるときに一気に切り替わらないよう間を繋ぐために、2.4.2 節で生成した形状を部分的に消去したモデルを使用する。高ディテールモデルと中ディテールモデルそれぞれに対して同じ半径を使用し形状を消去したモデルを、組み合わせて同時に表示する。図 2.13 は同じ半径を使用し形状を消去したモデル同士を組み合わせて表示している状態を図解したものである。

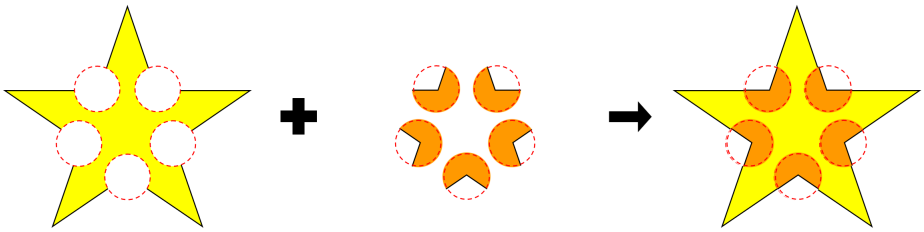
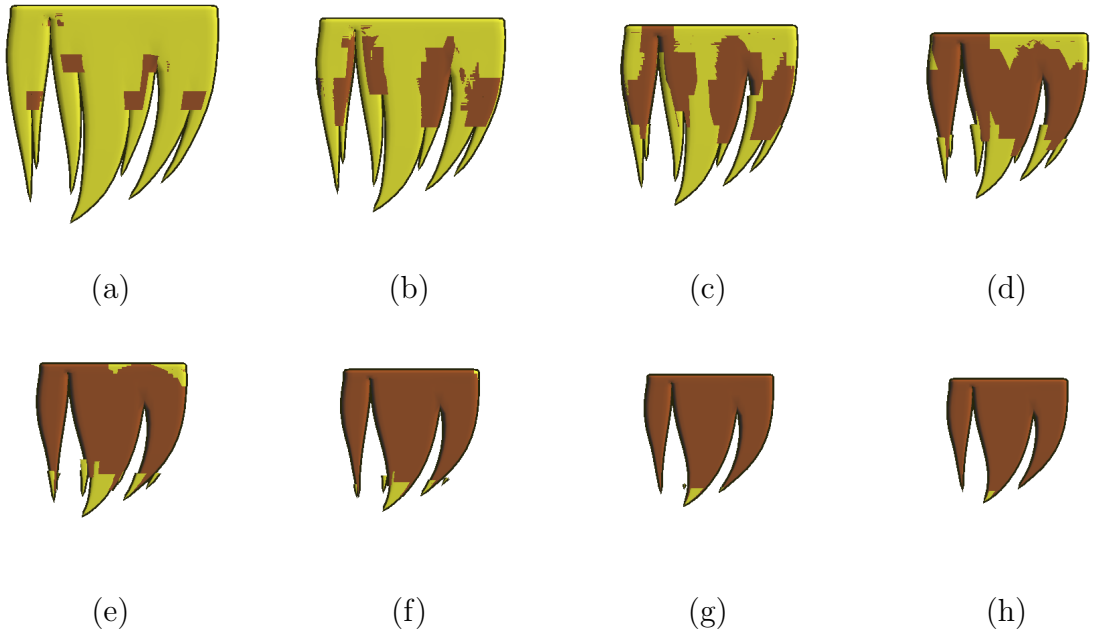


図 2.13: 2 種類のディテールのモデルを組み合わせて表示している状態

切り替える特徴点からの範囲の半径を (2.2) 式で求めた長さ l とし、モデルが視点から一定の距離離れたら、特徴点から半径 l の範囲が消去してある高ディテールモデルと、半径 l の範囲以外が消去してある中ディテールモデルを組み合わせて表示する。さらに視点からモデルまでの距離が離れたら、特徴点から半径 l の 2 倍の範囲が消去してある高ディテールモデルと、半径 l の 2 倍の範囲以外が消去してある中ディテールモデルを組み合わせて表示する。これを高ディテールモデルから中ディテールモデルに完全に切り替わるまで行う。また逆に、視点からモデルまでの距離が近づいた場合は上記の手順を逆に行い中ディテールモデルから高ディテールモデルに切り替える。図 2.14 は、数回に分けて部分的なディテールの変更行っている実行画面である。



(a) (b) (c) (d)

(e) (f) (g) (h)

図 2.14: モデル切り替え時の連続画像

図 2.14(a)〜(h) は順に高ディテールモデルから中ディテールモデルへ切り替えるときの、間を滑らかに繋いでいる連続画像であり、2 種類のディテールのモデルが組み合わさり同時に表示されている画像である。部分的に様子がわかりやすいように高ディテールモデルと中ディテールのモデルの色を変えてある。モデルのディテールを部分的に変更していっていることがわかる。

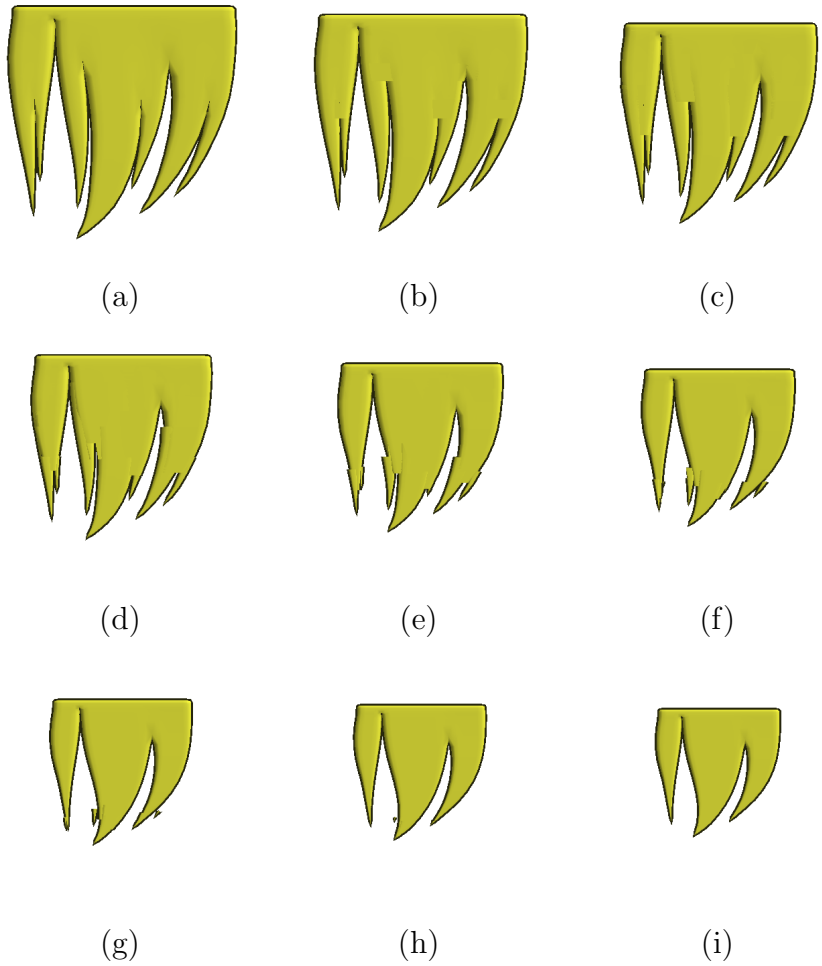
第 3 章

検証と考察

本章では、提案した 3DCG アニメーションにおける距離よる簡略化表現手法について検証と考察を行う。3.1 節では本手法を用いた場合、3DCG アニメーションで輪郭線が黒く潰れてしまうという問題が解決されいるかについて検証する。また、本手法はどのような時に効果的であり、どのようなときに効果的でないかについて述べる。3.2 節では本手法の問題点と今後の課題について考察する。

3.1 本手法の効果を検証

図 3.1 は本手法を用いた実行画面である。図 3.1(a) が視点からの距離が近く画面に大きく表示した高いディテールのモデル、図 3.1(i) が視点からの距離が遠く画面に小さく表示した低いディテールのモデルである。図 3.1(b)〜図 3.1(h) の画像は、2.4.3 節で述べた手法を用いて異なるディテールのモデルの間を繋いでいる画像である。



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

図 3.1: 本手法を実行したときの連続画像

2.1 節で述べたように、漫画や 2D アニメーションで用いる簡略化表現の特徴として以下の 2 点に注目した。

- モデルの輪郭線の太さが表示サイズにかかわらず一定である。
- 小さく表示したときも細かい部分の輪郭線が黒く潰れない。

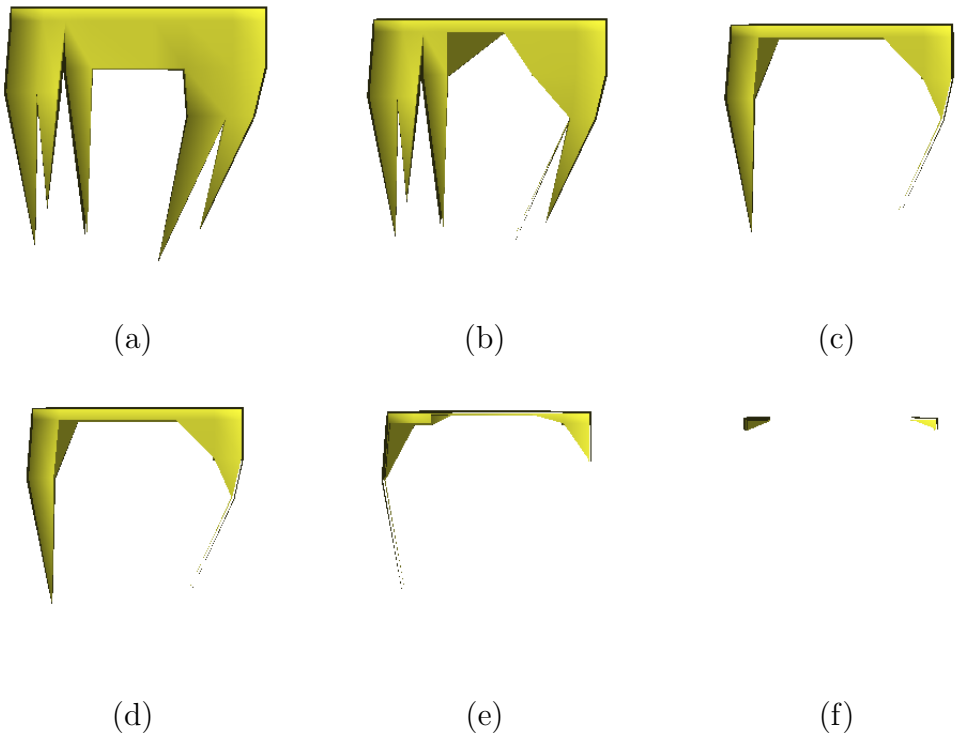
この 2 点については本手法によって実現することができているかどうか検証するために、本手法を適用したものと適用していないものを比較する。図 3.2 は様々な手法を用いて描画したモデルを比較したものである。

(a) (b) (c) (d)

図 3.2: 様々な手法を用いて描画したモデルの比較

図 3.2(a) は高いディテールのモデルを画面に大きく表示したもの、図 3.2(b) は特別な処理をせずにモデルの表示サイズを縮小したもの、図 3.2(c) は輪郭線のみ本手法を適用したもの、図 3.2(d) は本手法を全て適用したものである。図 3.2(b) については輪郭線の太さが変化してしまっているので漫画や 2D アニメーションのような表現とは言えない。図 3.2(c) は漫画や 2D アニメーションのような簡略化を行っていないので、細かい部分の形状がわかりづらい状態である。図 3.2(d) は輪郭線がの太さが一定であり、輪郭線が重なって潰れることも無く、上記の 2 つの条件を満たしていることがわかる。このように本手法は漫画や 2D アニメーションのような簡略化を実現し、3DCG の問題点を解決できたと言える。

続いて 2.4.3 節で述べた高いディテールモデルと低いディテールモデルの間を繋ぐ手法による、モデルを切り替えるときの滑らかさについての検証をする。図 3.1 で示したような例では、根元の凹形部分から少しずつ部分的に変更しており、滑らかに切り替わっていると言える。しかし、初めに用意したモデルの形状や状態によっては滑らかに切り替えることができない場合がある。図 3.3 にその例を示す。

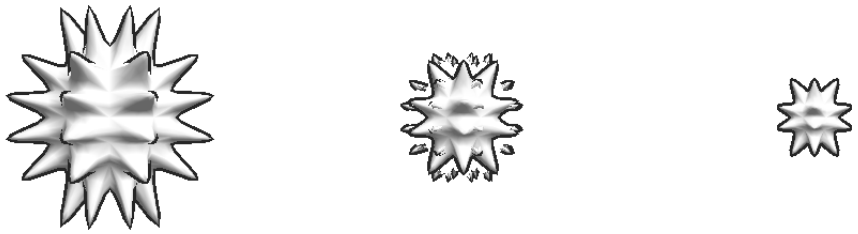


(a) (b) (c)

(d) (e) (f)

図 3.3: モデルのポリゴン数が極端に少ない例

図 3.3 はポリゴン数が極端に少ないモデルに対して、2.4.2 節で述べた手法を適用した 6 段階の部分的に形状が消去してあるモデルである。図 3.3(c)〜(f) では形状の変化が少なくなっていることがわかる。また (c) と (d) が同じ形状になっており、実際には 5 段階になってしまっている。本手法では部分的に形状が消去してあるモデルの形状生成をポリゴン単位で行っているので、図 3.3 のようにポリゴン数が極端に少ない場合は、部分的に形状が消去してあるモデルの段階数があまり多く作れず、滑らかな切り替えができなくなってしまう。



(a) (b) (c)

図 3.4: モデル同士の整合性を取らずに作成したモデルの例

図 3.4 は初めに用意しておく異なるディテールモデルを、2.3 節で述べたように異なるディテールのモデル同士の形状が重なるように制作していないものである。図 3.4 の例では、一定間隔で並んだ棘の数を減らしているだけなので、高いディテールのモデルと低いディテールのモデルの形状がうまく重ならなず、高いディテールのモデルの棘の中のいくつかが低いディテールの何もない部分に存在している。そのため、図 3.4(b) の切り替え時に高いディテールのモデルの棘の先端が宙に浮いているような状態になってしまっている。

3.2 問題点の考察

ここでは本手法の問題点について述べる。本手法の問題点について以下のような点を挙げる。

(a) ディテールの違うモデルを前もって作っておく必要があり、形状がディテールが高いモデルと低いモデルで対応していなければならない。

(b) ポリゴン数が少ないと、部分的に形状が消去してあるモデルをうまく生成できない。

(c) モデルの切り替えを滑らかにするためには、部分的に形状が消去してあるモデルの段階数を多く生成する必要がある。

(d) 輪郭線を描画するためのモデルを動的に変形しているので、描画処理が重くなる。

(a) は 2.2 節で述べたように、ディテールの違うモデルを作成する際に、ディテールの高いモデルと低いモデルの形状がある程度重なるように作成する必要がある。また、ディテールの段階を多く作るほど切り替えは滑らかに、制作者の意図通りに切り替えができるが、その反面制作時間が増加してしまうという問題点がある。この問題は本手法における最大の問題点であり、これを解決するためには低いディテールのモデルを自動生成できる手法を考えなければならない。動的に低いディテールのモデルを生成する LOD の技術 [17] は発展してきているので、今後これらの技術を応用することによって解決できる可能性はある。

(b) と (c) はのモデルを生成する際の問題で、滑らかに切り替えを行うためにはモデルの形状にかかわらずある程度のポリゴン数が必要となってしまう。これは描画処理が重くなる原因となる。

(d) は輪郭線を描画する処理に裏ポリゴンを利用する手法を用いており、輪郭線を一定の太さにするために、モデルの表示サイズが変わるたびに、輪郭線用のモデルの全頂点を法線方向に拡大縮小しているので、処理が非常に重い。この問題を解決するためには、拡大縮小するために利用する法線情報を全頂点から参照するのではなく、一部からのみ参照するなどの処理の軽減をするか、または輪郭線を描画する手法自体を変更する必要がある。本手法で取り入れた以外の輪郭線描画手法としては、Z バッファを調べて深度が不連続になっている部分に線を引く手法 [18] や、視線と平行な面に黒く色を付ける手法 [19] などがある。

第 4 章

まとめ

最後に、本研究のまとめと今後の展望について述べる。

本研究では、3DCG アニメーションにおいて漫画や 2D アニメーションのような輪郭線の簡略化表現を実現した。漫画や 2D アニメーションの簡略化表現を 3DCG アニメーションに取り入れることによって、3DCG アニメーションにおいてモデルの表示サイズを縮小したときに起こる輪郭線が重なって黒く潰れるという問題を解決することができた。しかし、3.2 節で述べたようにいくつかの問題点も残っている。モデル作成の手間が必須であることがその問題点の１つであり、この問題は切り替えていくモデルを自動生成する手法を提案する必要がある。LOD を行う際の低いディテールのモデルを自動生成する研究 [20] も行われており、それらの手法を取り入れることにより実現できる可能性がある。その場合は一般的な LOD のようにただ特徴を残すだけではなく、漫画や 2D アニメーションのように特徴がよりわかりやすくなるように形状を変形させる簡略化を行ったモデルを自動生成しなければならない。また本手法をリアルタイムなコンテンツに使用する場合は描画処理の効率化も必要となる。

本研究ではモデルの輪郭線にのみ注目したが、3D モデルは形状のみではなくモデルの面に貼ったテクスチャによっても表現される。テクスチャで表現する顔のパーツなども 1.1 節で述べたように漫画や 2D アニメーションでは簡略化している。画面上の表示サイズによってテクスチャを切り替える技術としては MIPMAP とい

うものがあり、今後 MIPMAP の技術を組み合わせることによって表現の幅が広がるのではないかと考えられる。

謝辞

本研究を進めるにあたり、多くのご指導をいただきました東京工科大学メディア学部の渡辺大地講師とその他多くの講師の方々、院生の方々、また研究室で共に戦い意見を交し合ったゲームサイエンスプロジェクトのメンバーに感謝いたします。

参考文献

[1] Aaron Hertzmann, Denis Zorin, “ Illustrating Smooth Surfaces ”, SIGGRAPH2000, 2000.

[2] Robert D. Kalnins, Lee Markosian, Barbara J. Meier, Michael A. Kowalski, Joseph C. Lee, Philip L. Davidson, Matthew Webb, John F. Hughes, Adam Finkelstein, “ WYSIWYG NPR: Drawing Strokes Directly on 3D Models ”, SIGGRAPH2002, 2002.

[3] Amy Gooch, Bruce Gooch, Peter Shirley, Elaine Cohen, “ A Non-Photorealistic Lighting Model for Automatic Technical Illustration ”, SIGGRAPH98, 1998.

[4] 村上恭子, 鶴野玲治, “ 顔料及び支持体の特性を考慮したパステル画風レンダリング ”, 芸術科学会論文, 2002.

[5] 中川大介, 藤本忠博, 村岡一信, 千葉則茂, “ 水彩パース図風レンダリング手法 ”, 芸術科学会論文, 2003.

[6] 「APPLESEED」, 士郎正宗, 青心社, APPLESEED フィルムパートナーズ, 2004.

[7] 「FREEDOM」, サンライズ, 2006.

[8] 古野泰, “ 3DCG における漫画的なスピード誇張表現に関する研究 ”, 東京工科大学大学院メディア学研究科修士論文, 2006.

[9] Tomas Akenine-Moller, Eric Haines, 「リアルタイムレンダリング第 2 版」, ボーンデジタル, 2006.

[10] 橋本洋志, 小林裕之, 図解 OpenGL による 3 次元 CG アニメーション, 旺文社, 2005.

[11] 吉田典正, 北嶋克寛, “ ポリゴンモデルを対象とする視界に依存する適応的表示 ”, 電子情報通信学会論文誌, 2000.

[12] Osamu Mizuno, metasequoia,

<http://www.metaseq.net/>.

[13] 渡辺大地, FK Tool Kit System,

<http://www.media.teu.ac.jp/~earth/FK/>.

[14] Hugues Hoppe, “ Progressive Meshes ”, SIGGRAPH96, 1996.

[15] Hugues Hoppe, “ View-dependent refinement of progres-sive meshes ”, SIGGRAPH97, 1997.

[16] 鈴木隼人, “ リアルタイム 3DCG における米国漫画調レンダリング手法 ”, 東京工科大学大学院メディア学研究科修士論文, 2004.

[17] トライゼット西川善司, “ GDC2003 EXPO レポート ”,

<http://www.4gamer.net/specials/gdc24/cry.html>.

[18] 今給黎 隆, ”T-Pot ”,

< http://www.t-pot.com/>.

[19] マイクロソフト, “ 頂点シェーダを使用したDirectX 8でのトゥーンレンダリング ”,

<http://www.microsoft.com/japan/msdn/directx/techart/DXVertex.aspx>.

[20] Michael Garland, Paul S. Heckbert, “ Surface Simplification Using Quadric Error Metrics ”, SIGGRAPH97, 1997.