

2006年度 卒業論文

3DCGにおける漫画的な  
スピード誇張表現に関する研究

指導教員：渡辺 大地講師

メディア学部 ゲームサイエンスプロジェクト

学籍番号 M0103376

古野 泰

Media Science

2006年度 卒業論文概要

論文題目

3DCGにおける漫画的な  
スピード誇張表現に関する研究

メディア学部

学籍番号：M0103376

氏名

古野 泰

指導  
教員

渡辺 大地講師

キーワード

3DCG、ノンフォトリリスティック、漫画、誇張表現  
モーションブラー、効果線

近年、3DCGにおけるノンフォトリリスティックな表現は盛んに行なわれている。ノンフォトリリスティックな表現の普及に伴い、漫画的な表現が3DCGのTVゲームやアニメーションにおいて多く見られるようになった。3DCGにおける漫画的な表現として現在使われている手法は、輪郭線の描画、セル画的な2~3階調の色表現、漫画の描き文字表現、効果線の描画、影にハッチングのような斜線を描画、などが挙げられる。しかし、現在も実現されていない漫画的な表現は数多くある。そこで本研究では、現在の3DCGにおけるノンフォトリリスティックな表現において、実現されていない漫画的なスピード誇張表現に注目し、その表現を3DCG上で再現することで、3DCG上においても漫画的なスピード誇張表現が効果的かどうかを検証した。そこで注目したスピード誇張表現が、キャラクターの境界線のブレによるスピード誇張である。これを「境界線ブラー」と定義し、3DCGにおける再現を試みた。3DCGにおいて「境界線ブラー」を実現するため、境界線のモデルを変形させる方法と、パーティクルを使用する方法を用いた。

効果を検証した結果、キャラクターの移動の速度が速い場合、その効果は従来の効果線による表現と大きな違いは無かった。しかしスローモーションなど、あえて動きを遅くする演出の場合に、この表現は動線のランダム性の強調によるスピード誇張の効果と、キャラクターの境界線の形が曖昧に表現されることによるスピード誇張という効果が得られた。本研究の結果としては、パーティクルを使用する方法のほうが、境界線のモデルを変形させる方法より、漫画の境界線のブレ表現を再現できた。境界線のモデルを変形させる方法は、視点の変更に対応できない、境界の曖昧さが表現しきれしていない、動線の曲線表現に手間がかかる、という問題がある。しかしパーティクルを使用する方法についても、陰面に隠れるはずの動線も表示されてしまう、モデルのブレが移動の前面にも表示されてしまい動線の前にはみ出す、パーティクルを大量に描画するため処理が重い、という問題があり、これらの点では境界線のモデルを変形させる方法のほうが優っているといえる。これらの問題は、今後の課題として解決すべき点である。

# 目次

第1章	はじめに	1
第2章	漫画的スピード誇張表現	3
2.1	漫画における境界線のブレによるスピード誇張表現	3
2.2	各表現形態の基本となる実写におけるブレとは	5
2.3	アニメーションにおけるブレ表現、オバケ	7
2.4	3DCGにおけるブレ表現、モーションブラー	8
第3章	境界線ブラーの実現手法の提案	10
3.1	既存のセルシェーディング手法	10
3.2	境界線のモデルを変形させる方法	13
3.3	パーティクルを使用する方法	15
3.3.1	先行研究	15
3.3.2	パーティクルを用いた境界線ブラーの実現方法	19
第4章	検証と考察	26
4.1	漫画的なスピード誇張表現として効果的かどうかの検証	27
4.1.1	境界線のモデルを変形する方法について	27
4.1.2	パーティクルを使用する方法について	30
4.2	効率化に関する検証	33
4.3	問題点の考察	34
第5章	まとめ	36
	謝辞	37
	参考文献	38

# 第 1 章

## はじめに

近年、油絵調や鉛筆画調、セル画調などの人の手によって描かれている画像を、3DCG を用いて再現するノンフォトリリスティックな表現の研究 [1][2][3][4] が盛んに行われている。ノンフォトリリスティックな表現の普及に伴い、漫画的な表現が 3DCG の TV ゲームやアニメーションで多く見られるようになった。主に輪郭線の描画とセル画的な 2~3 階調の色表現が主流で、他には漫画の描き文字表現や効果線の描画、影にハッチングのような斜線を描画などの表現が使われているものもある。このように多くの漫画的な表現が 3DCG において再現されているが、現在も 3DCG において実現されていない漫画的な表現は数多くある。本研究では、現在も 3DCG において実現されていない、境界線のブレによるスピード誇張表現というものを 3DCG において実現し、その表現が 3DCG においてもスピード誇張表現として効果的かどうか、その検証を行った。

漫画における境界線のブレによるスピード誇張表現とは、キャラクターが高速移動する際、キャラクターを構成する境界線を、移動の後方にブレているように線を描くことによって、キャラクターが高速で移動していることを表現するものである [5]。また、移動の後方に線を描くことで、止まり絵である漫画の 1 コマの中に、動きの方向性を示すという効果がある。この表現は、コマという止まり絵の中に、スピードや動きを感じさせるための、非写実的な誇張表現であるといえる。

このような非写実的なスピード誇張表現に関する研究には、大林らの”3D アニ

メーションのためのカットーンブラー”[6]がある。この研究では、2Dアニメーションにおける非写実的な動きの効果を「線」、「残像」、「ゆがみ」の3種類に分類し、それらの表現を3DCG上で実現している。以下の図1.1、図1.2、図1.3に実装した画像を示す。



図 1.1: 「線」による効果



図 1.2: 「残像」による効果



図 1.3: 「ゆがみ」による効果

この研究によって、本研究に近い表現である「線」や「ゆがみ」による非写実的な動きの効果を3DCG上で実現しているが、境界線のブレによるスピード誇張表現自体は実現されていない。その他の非写実的なスピード誇張表現に関する研究[7][8]や、3DCGのアニメーションやゲームにおいても、この境界線のブレによるスピード誇張表現は実現されていない。そこで本研究では、境界線のモデルを変形させる方法とパーティクルを使用する方法を用いて、境界線のブレによるスピード誇張表現を3DCG上で実現する手法を提案する。

## 第 2 章

# 漫画的スピード誇張表現

本章では、本研究が注目した漫画的なスピード誇張表現がどういったものであるかを述べる。2.1 節では、漫画における境界線のブレによるスピード誇張表現について述べる。2.2 節以降では、漫画におけるブレ表現を 3DCG において実現する上で、アニメーションや実写映像などの各表現形態におけるブレ表現の調査結果を述べる。2.2 節では実写におけるブレについて、2.3 節では 2D のアニメーションにおけるブレ表現について、2.4 節では 3DCG のアニメーションにおけるブレ表現について述べる。

### 2.1 漫画における境界線のブレによるスピード誇張表現

本研究において注目した漫画的な表現は、キャラクターの境界線のブレによるスピード誇張表現というものである。漫画的な表現を実現する上で、線表現は欠かせないものである [9]。白い紙に黒いインクで描かれる漫画の絵においては、モノクロで色が存在しないので、物体は線で境界を描くことで表現する。漫画の絵において物体を構成している線は、非常に重要であるといえる。その線表現のなかで、現在の 3DCG において実現されていない表現が、このキャラクターの境界線のブレによるスピード誇張表現である。以下の図 2.1 にその表現を示す。



図 2.1: 境界線のブレによるスピード誇張表現の例 (山口貴由, 南條範夫, 秋田書店)

図 2.1 は、山口貴由著の「シグルイ」[10] という作品である。この図の刀の部分において使用されているのが、マンガにおいて見ることが出来るスピード誇張表現である。白い紙に黒いペンで表現するマンガという媒体においては、白黒に分かれたはっきりとした画面が基本であり、写真のブレのようなぼやけた表現を再現するには、線で表現するための何らかの工夫をしなければならない。そのようなブレを線で表現したのものが、この図に示しているものである。ここよりこの表現のことを「境界線ブラー」と呼称する。また、この表現に用いられる移動方向に引かれた線のことを「動線」と呼ぶ[11]。この表現は、境界をブレたように描き、形を曖昧にする事によって、認識できないほどのスピードというものを表現しているといえる。また、物体が移動した方向に線が引かれることによって、物体の移動の方向性を示し、線の長さはそのコマの時間における物体の移動幅を表

している。線の長さ分、このコマでは物体が移動した、ということである。1コマという瞬間における移動幅を強調することで、瞬間にこれだけ動いたというスピード誇張の役割をしている。さらに、この表現では、動線の伸びの長い部分と短い部分がはっきりと分けて描かれている。このように、線の長さによってランダム性があることによって、その物体が移動した幅がはっきりとわからない。これは、移動した幅を曖昧にすることで、先に述べたような認識できないほどのスピードである、ということを強調している。従来の効果線表現にも線の長さのランダム性はあるが、この表現のようにはっきりと長い部分、短い部分と分けて描かれてはいない。つまりこの表現は、線のランダム性によるスピード誇張をより強調した表現であるといえる。まとめると、この表現には以下のような効果がある。

- 境界の曖昧さによるスピードの誇張
- 線による移動方向の強調
- 線の長さによる移動幅の強調
- 線のランダム性を強調したスピードの誇張

これらの中で、この表現の特徴として重要なのが、境界の曖昧さと線のランダム性を強調している点である。このふたつの表現が、現在3DCGにおいて実現されていないブレ表現の要素であるといえる。以上のような表現を、3DCGにおいて実現する。

## 2.2 各表現形態の基本となる実写におけるブレとは

実写映像などにおいて、高速で移動する物体のブレというものは、撮影する際のシャッタースピードが遅いことが原因で起こる。シャッターが開いている間はずっとフィルムが感光しているため、その間移動している物体は移動した分だけ伸びたように撮影される。こうすることで1コマ1コマがブレることになるが、アニメーションにした場合、フレーム間の中間時間を補完する作用となり、結果とし



てアニメーションを滑らかに見せる効果がある。以下の図 2.2 に、実際のブレている様子を示す。

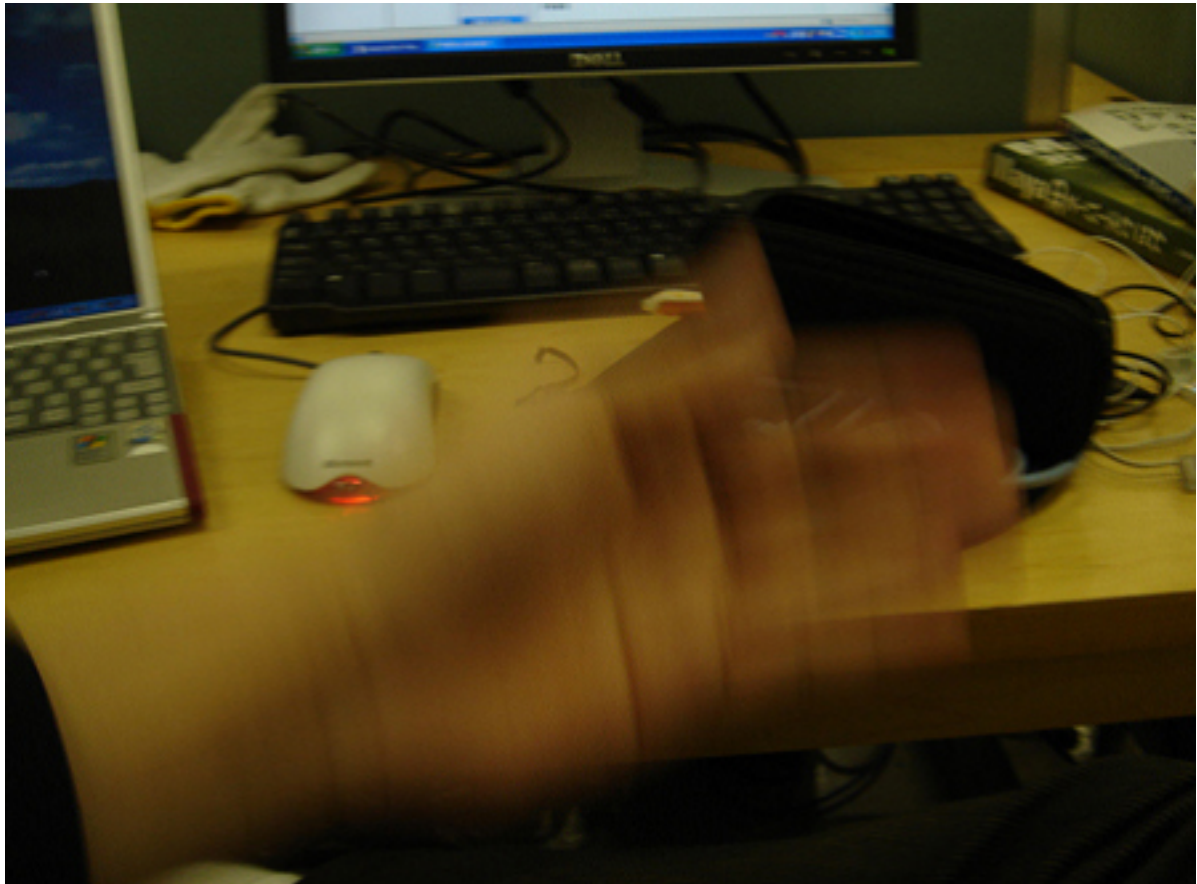


図 2.2: 実写におけるブレ

また、映像とは別に、人間が物を見るときに起こる現象で、高速で移動する物体に対して動体視力が追いつかないため、その物体の移動した跡が残像として見える、というものがある。この現象も、漫画やアニメーションなどの表現形態において、物体が速く動いていることを示すために用いられる。

以上の二つの現象は、他の表現形態におけるブレ表現の基本となるものである。各表現形態のブレ表現は、以上に述べた現象から派生したものであると考えられる。

## 2.3 アニメーションにおけるブレ表現、オバケ

アニメーションにおけるブレ表現は2種類ある。移動物体の形を歪ませたり引き伸ばしたりする「オバケ」という表現と、漫画にも用いられている境界線のブレ表現である。

まず「オバケ」という表現に関して述べる。以下の図 2.3 にその表現を示す。

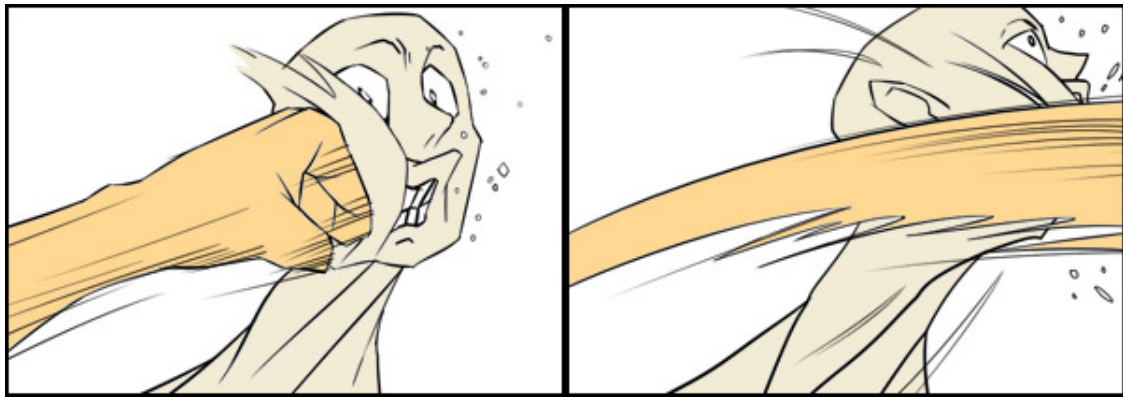


図 2.3: 2D アニメーションにおけるブレ・オバケ

図 2.3 は、パンチを打つアニメーションを表している。左図がパンチが当たった瞬間、右図がパンチを打ち抜いた瞬間の図であるが、右図の腕の部分引き伸ばされて形がなくなっている。このように移動物体の形を歪ませたり引き伸ばしたりして、実写映像における残像の現象を、2D のアニメーションで表現している。これが 2D アニメーションにおける「オバケ」という表現 [12][13] である。残像現象におけるブレを、物体の境界が線で描かれるために、物体と物体の境目がはっきりとしてしまう 2D アニメーションにおいて表現するため、物体の形自体を歪ませるという方法を編み出したと考えられる。漫画と違う点は、2D アニメーションには漫画には無い色という概念が存在しているため、色を使うことで物体の面を表現することが出来る。漫画における境界線のブレは、色という概念が無いために線をブレさせるだけでこの「オバケ」の効果を得られるものである。アニメの場合において、境界線のみ動線にしたとしても、色が存在することで面の存在は

曖昧にならず、はっきりと存在したままになってしまうので、漫画のような動線のみによるブレ表現は2Dアニメーションには適さない。2Dアニメーションにおいては、「オバケ」表現と動線表現は組み合わせて使われることが多い。アニメにおいて境界線のブレは、スピード誇張に使用するというよりは、荒々しさや力強さなどを出すために使われていると考えられる。

ただし、漫画の原作のあるアニメーション作品の場合、原作の漫画において境界線のブレがスピード誇張として印象的に使われているシーンでは、あえて同じように境界線のブレを使用するという場合もある。以下の図2.4にその表現の例を示す。



図2.4: 2Dアニメーションにおける境界線のブレ表現

図2.4のキャラクターの腕の部分においては、面の存在は消えて線だけになっている。このような線表現も、原作の漫画のシーンを再現するために、2Dアニメーションに使用されることがある。

## 2.4 3DCGにおけるブレ表現、モーションブラー

3DCGにおけるブレ表現は、モーションブラーと呼ばれ、写実的なものと、非写実的なものの2つが挙げられる。写実的なブレ表現は、実写において見られるブレの現象を再現しようとしたものである。すなわち、シャッタースピードの遅れ

によるブレと、残像効果の2つである。シャッタースピードの遅れによるブレは、2.2節で述べたように、アニメーションを滑らかに見せる効果を狙ったものである。シャッタースピードの遅れによるブレを3DCGにおいて再現する方法として、アニメーションのレンダリングの際に、前フレームと現在のフレーム間の画像を任意の数描画し、描画した画像を平均化するという方法がある[14][15]。こうすることで、フレーム間の描画数分引き伸ばされた画像になり、アニメーションにしたときに滑らかな動画になる。これが、シャッタースピードの遅れによるブレの原理に基づいた表現方法である。ただし、この方法は描画速度が重くなってしまいうという欠点がある。

残像効果を再現したブレは、2.2節で述べたように、速く動いていることを示すスピード誇張表現として用いられる。その表現方法は、3Dアニメーションのレンダリングの際に、前フレームの画像を薄く描画して残す、というものである[16][17]。前のフレームであればあるほど薄くしていく。こうすることで、擬似的に実写における残像のような効果が得られる。この方法は、処理も簡単で描画速度もそれほど重くないので、リアルタイム3DCGに多く用いられる。

3DCGにおける非写実的なブレ表現は、漫画や2Dアニメーションなどの表現を再現しようとしたものである。2.3節で述べたような「オバケ」表現や、2Dアニメーション的な残像表現が挙げられる。これは、第一章で述べたカトゥーンブラー等の研究において実現されている。写実的なモーシヨンプラーに比べ、現在の3DのTVゲームやアニメーションにおいてはあまり使用されていない。

## 第 3 章

# 境界線ブラーの実現手法の提案

本章では、本研究が提案する境界線ブラーの実現手法について述べる。本研究において境界線ブラーの動画の制作は、Maya と After Effects を使用した。前提として、セルシェーディングで制作したモデルを使用する。そこで 3.1 節で既存のセルシェーディング手法について説明した後、3.2 節では境界線のモデルを変形させる方法について、3.3 節ではパーティクルを使用する方法について述べる。

### 3.1 既存のセルシェーディング手法

本手法の適応する際の条件として、セルシェーディングを用いて制作したモデルに適応することを想定している。これは、境界線ブラーが非写実的な効果であるので、非写実的な映像を制作する場合に使うという考えからのことである。そこでまずセルシェーディングについて説明する。セルシェーディングとは 3DCG をセル画調にレンダリングする技術であり、当初は手描きでは困難な、物体の回転や視点の移動を伴った滑らかに動くセル画調のアニメーションを 3DCG を用いて容易に作成すること、及び、3DCG を利用する事により、絵を毎コマ描かずに済ませるといった風に、作画作業の手間を省くといった目的から開発された手法である。

本研究においては maya 上でこのセルシェーディングを行った。その方法は、

maya3D スーパーテクニク [18] において紹介されているランプシェーダを使用する方法である。その方法の手順は、以下のとおりである。

1. モデルに適応するマテリアルの設定をランプシェーダに設定
2. グラデーションの設定で色を 2 階調に指定
3. ランプの広がり方を brightness に設定
4. スペキュラの明るさの値を 0 に設定

以上の手順を行うことで、以下のようなセルシェーディングの画像 3.1 を得た。

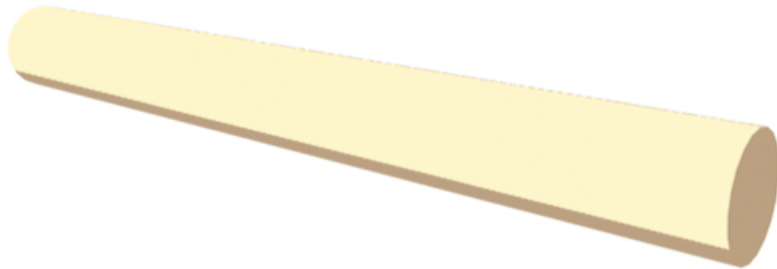


図 3.1: セルシェーディングの画像

さらに、セルシェーディングにおいて行われる表現として、境界線の描画がある。漫画や 2D アニメーションにおいては、物体と物体の境目に境界線が描かれる。その境界線を、3DCG においても描画することで、よりセル画調に近い画像が得られる。その方法として、“米国マンガ調セルシェーディングに関する研究” [19] において述べられている、3D モデルを引き伸ばす方法を用いた。以下はその論文からの引用である。

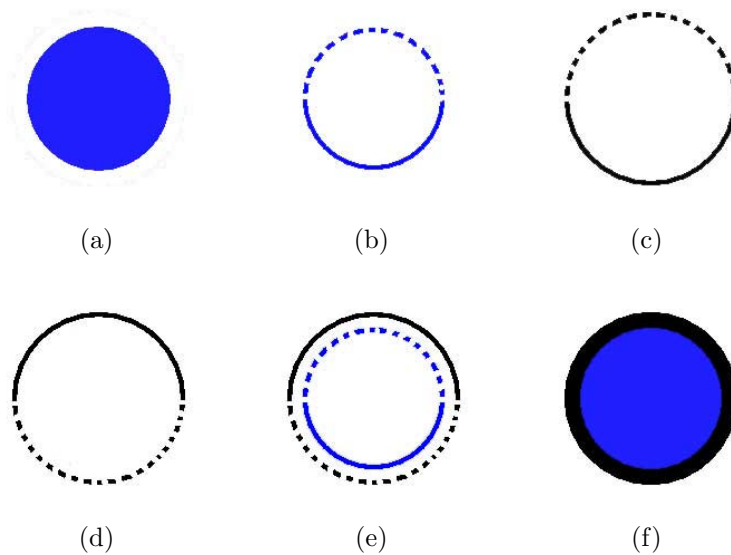


図 3.2: モデルを引き伸ばす輪郭線描画法

球体のモデルを用いて説明する。図 3.2(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 3.2(b) である。図 3.2(b) ~ 3.2(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。輪郭線を描く処理の行程は

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。
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。

である。図 3.2(c) ~ 3.2(e) はこの処理を図示した物であり、図 3.2(c) は拡大し面を黒くした図、図 3.2(d) は面の表裏を反転させた図、図 3.2(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 3.2(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 3.2(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、以下のような画像 3.3 を得た。

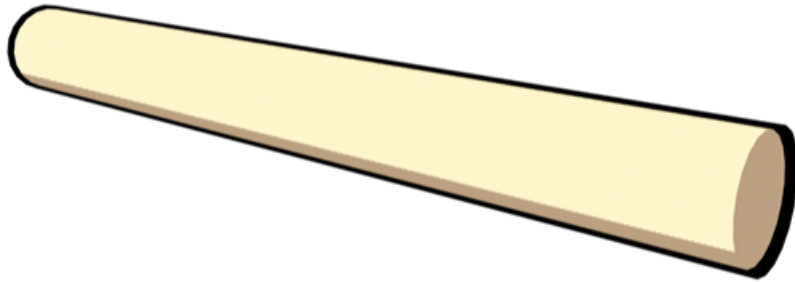


図 3.3: 輪郭線を描画した画像

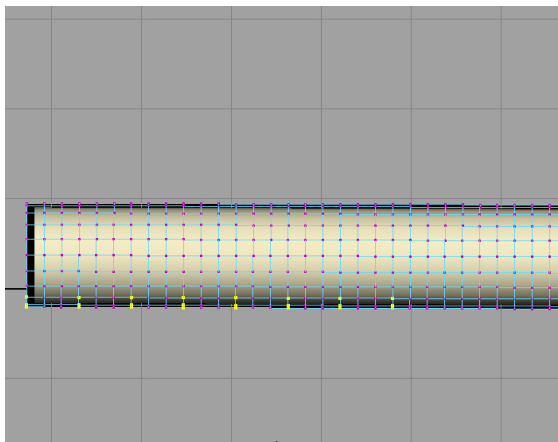
このセルシェーディングの手法を基本として、境界線ブラーの効果を加える。

## 3.2 境界線のモデルを変形させる方法

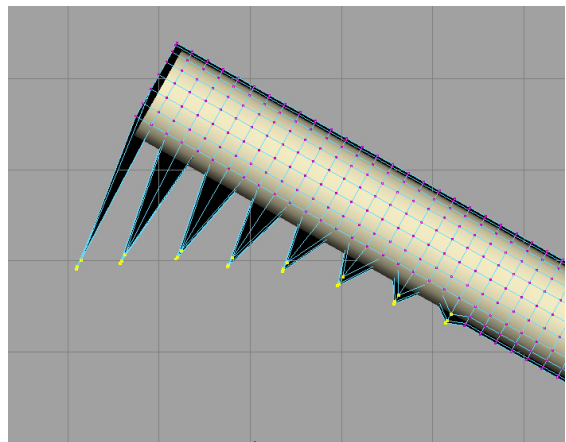
ここでは、境界線のモデルを変形させる方法について説明する。境界線を反転モデルを使って表現する手法において、境界線を表している反転モデルを変形させて動線を表示する。

カメラの位置を決定してから、境界線のモデルにアニメーションをつける。境界線のモデルの、引き伸ばしたい頂点を選択し、選択した頂点を引き伸ばすアニメーションをつける。この際に、引き伸ばしたい頂点の間の頂点をモデルの内側に移動させるアニメーションをつける。以下の図 3.4 にその状態を示す。





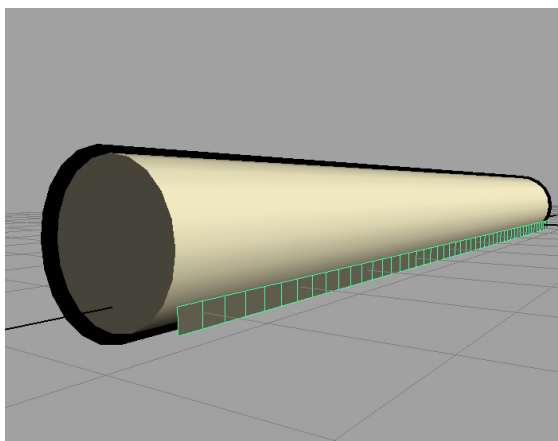
(a) 境界線モデルの頂点の選択



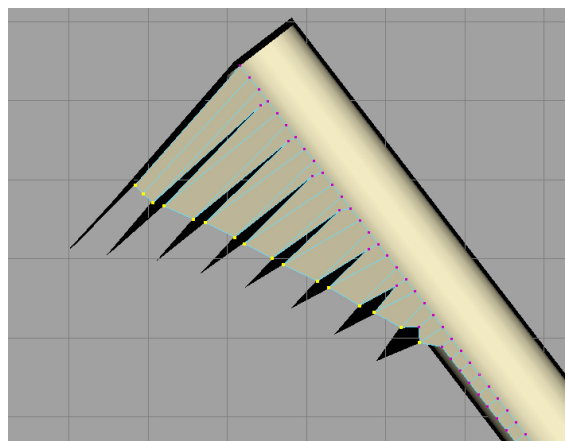
(b) 引き伸ばすアニメーション

図 3.4: 境界線のモデルの変形

図 3.4(a) のように境界線のモデルの頂点を選択し、図 3.4(b) のように移動の後方に境界線のモデルを引き伸ばすアニメーションをつける。この状態に、さらにモデルの色が動線と動線の間を描かれることを再現するために、移動の後方の境界線に当たる部分に、モデルの色と同じ色の板ポリゴンを配置する。この板ポリゴンを、境界線のモデルの変形に合わせてアニメーションさせる。以下の図 3.5 にその状態を示す。



(a) 板ポリゴンの配置



(b) 板ポリゴンのアニメーション

図 3.5: 動線と動線の間モデルの色を表現

図 3.5(a) のように、カメラに対して移動の後方の境界線が隠れるような位置に、モデルの色と同じ色に設定した板ポリゴンを配置する。そして図 3.5(b) のように、動線のアニメーションに合わせて板ポリゴンにアニメーションをつける。こうすることで、モデルの境界が曖昧になっていることを強調する。以上のような処理を行い、境界線ブラーの動画を制作した。

### 3.3 パーティクルを使用する方法

ここではパーティクルを使用した境界線ブラーの表現方法について説明する。境界線ブラーにおける動線を、パーティクルの点を用いた方法で表現し、モデルの境界の曖昧さを Maya のモーションブラーの機能を使って表現する。そうして表現した動線とモデルの画像を最終的に合成する、という方法である。

#### 3.3.1 先行研究

パーティクルを使用した線表現の先行研究として、中村陽介の”3DCG アニメーションにおける効果線表現手法の提案と効率化に関する研究” [7] がある。以下にその内容を引用する。

物体に対しての速度の効果線とは物体の通過した軌跡を線によって描いたものである。3DCG で物体の通過した軌跡を表現するためにパーティクルを使用する。

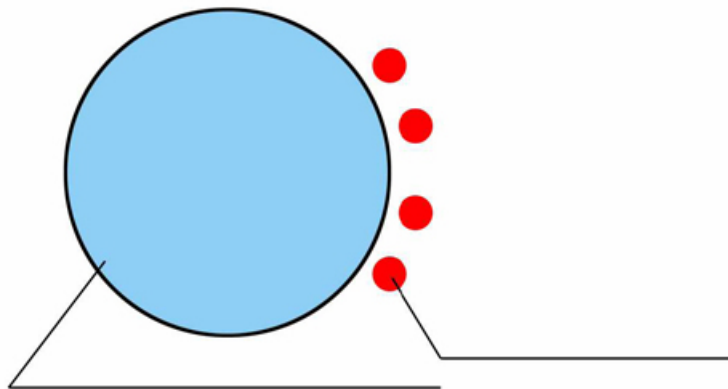


図 3.6: オブジェクトにパーティクル発生源を付加

図 3.6 のように、効果線が発生させたい任意の場所に点ポリゴンによるパーティクルの発生源を設置する。パーティクルの粒子は重力、風等の影響を受けず、発生したその場所に静止する。

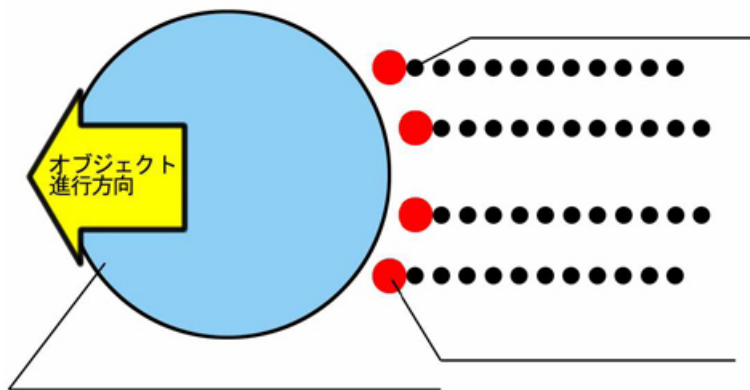


図 3.7: パーティクル発生源よりパーティクル発生

図 3.7 のように、オブジェクトを移動させるとパーティクル発生源の通過した軌跡にパーティクルが設置される。パーティクル間の隙間が小さいほど、滑らかな効果線が得られる。描画される効果線の長さはオブジェクトの速度によって変化するが、速度毎の効果線の最高長はパーティクルのエイジによって決まる。

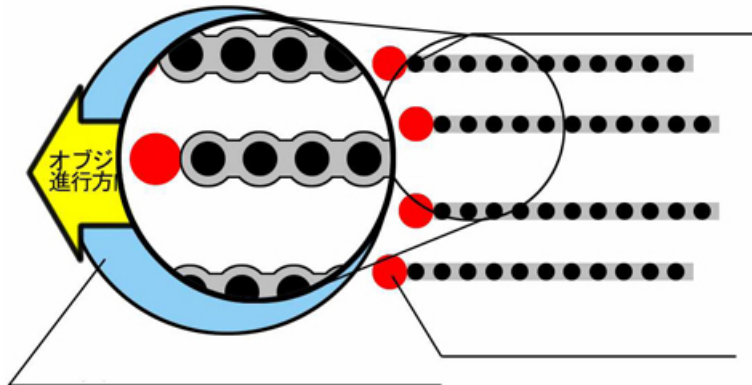


図 3.8: 発生したパーティクルを HyperVoxels によりライン化

図 3.8 のように、3DCG 化を行うために使用した 3DCG 制作ツール Light-Wave3D の標準機能である HyperVoxels (ボリュームレンダリング効果の一つ) を、パーティクル発生源の軌跡に配置されたパーティクルに適用しライン化する。パーティクルに適用する HyperVoxels サイズ設定によって効果線の太さは決定する。



図 3.9: パーティクルエイジと HyperVoxels 透明度的変化

図 3.9 のようにパーティクルのエイジに対して HyperVoxels の透明度を適用することによって効果線の頭と尾を滑らかに消すことができる。

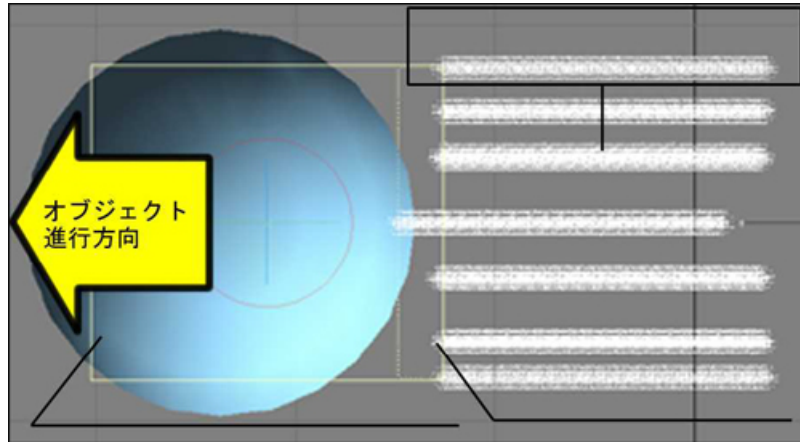


図 3.10: LightWave3D 上で設定したオブジェクト

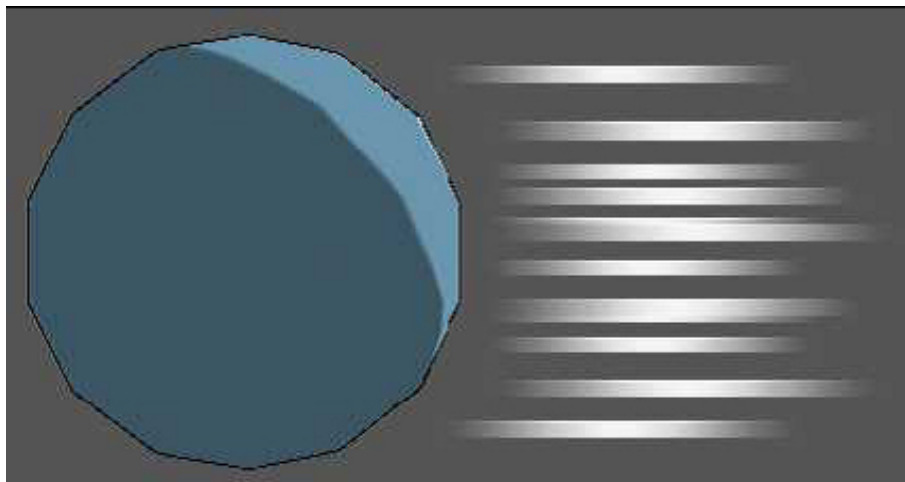


図 3.11: 図 3.8 のレンダリング画像

図 3.10 は実際に LightWave3D 上で設定した画像で、以上の処理を行って、レンダリングした画像が図 3.11 である。以上が、パーティクルによって効果線を描く既存手法である。

### 3.3.2 パーティクルを用いた境界線ブラーの実現方法

既存手法を元に、境界線ブラーの3DCG上での再現を行う。まず、より漫画の表現のイメージがわかりやすいようにキャラクターモデルを使用して再現する。以下の図3.12が研究に使用したモデルである。

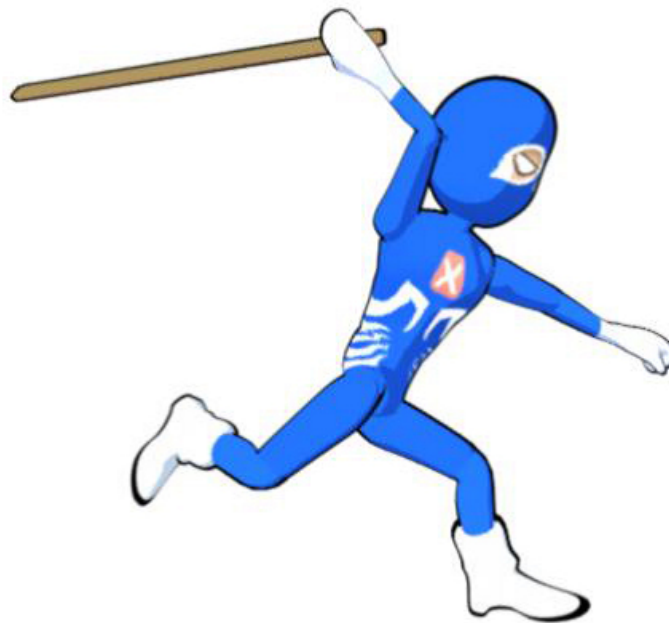
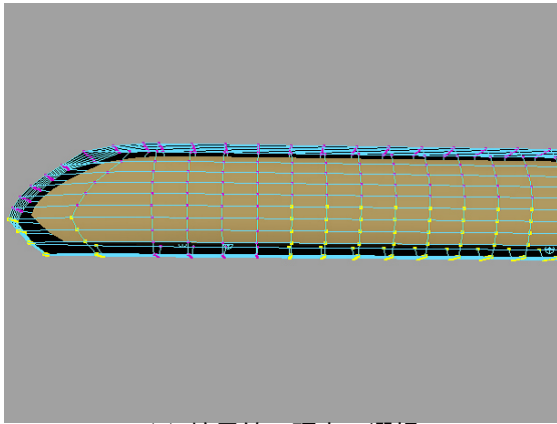
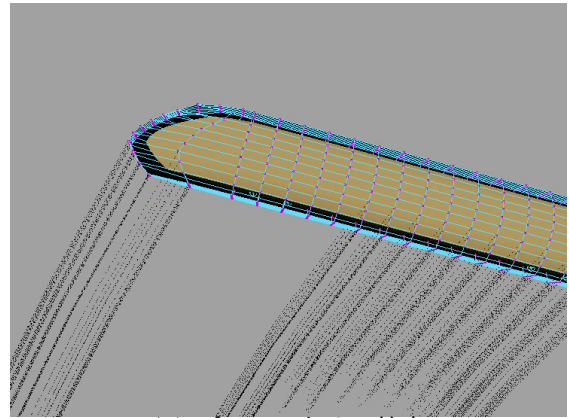


図3.12: 用意したセルシェーディングのモデル

このモデルにパーティクルで動線を付加する。この方法の手順としては、まず線を出したい境界線の部分を選択する。線を表示したくない部分、線の抜けの部分は選択しない。そこで、「particles」タブの「emit from object」を選ぶことで、モデルの選択した頂点からパーティクルを放出することが出来る。パーティクルのレンダータイプは points に設定し、点として表現する。点のポイントサイズは最小に設定した。また、エミッターのタイプは directional にして、パーティクルが一定の方向に連続して放出されるようにする。放出のスピードは0とする。スピードを0にすることでモデルが移動した軌跡に点が残り、物体の後方に線が描かれたように見える。以下の図3.13に、その画像を示す。



(a) 境界線の頂点の選択



(b) パーティクルの放出

図 3.13: モデルの頂点からのパーティクルの放出

図 3.13(a) が、境界線のモデルの動線を描画したい頂点を選択した図であり、図 3.13(b) が、選択した頂点からパーティクルが放出されている図である。次に、線の長さをグループに分け、それぞれのグループでパーティクルの設定を変えて線の長さの違いを表現する。以下の図 3.14 にその様子を示す。

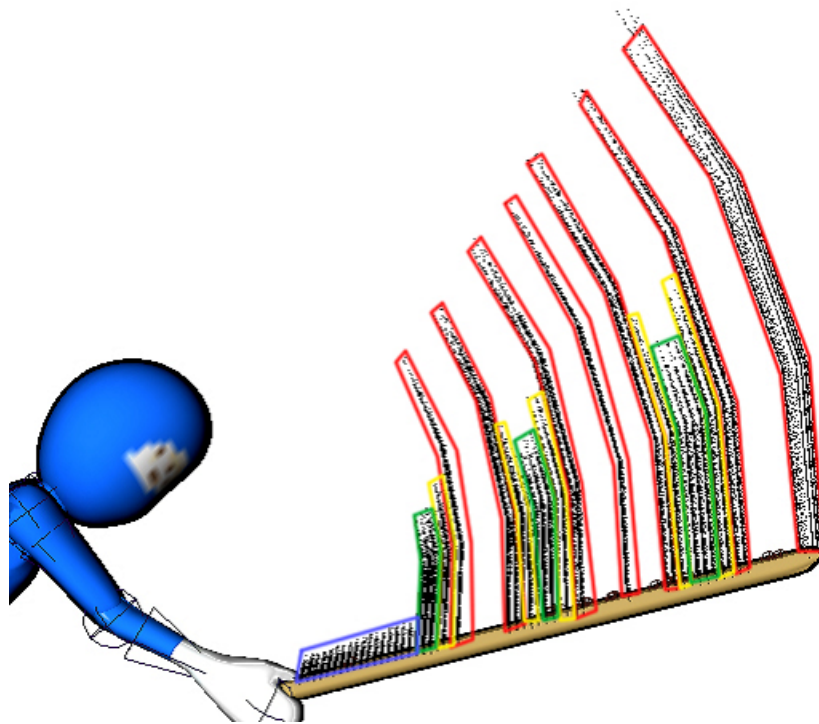


図 3.14: 動線の長さのグループ分け

この場合では、最も線が長い部分(赤)、次に長い部分(黄)、短い部分(緑)、最も短い部分(青)の4種類に分けて設定した。パーティクルの点の連続による線表現は、1秒間に放出する粒子の数(Rate)を高くすることによって、より線らしくなる。ただし、Rateの数値を高くしすぎると、動作が重くなってしまっているので、調整が必要である。本研究ではRateの数値を1000に設定した。動線の線の長さは、パーティクルの寿命によって決まる。本研究の場合では、寿命の値であるLifespanを、最も長い線で0.100、次に長い線を0.060、短い線を0.050、最も短い線を0.010に設定した。この値は、モデルの移動スピードによっても長さが変わってくるので、状況に応じて調整が必要である。さらにlifespan modeをrandom rangeにして、ランダムな値を0.010に設定した。こうすることで、線の後方部分の点がある程度ランダムに消えていくので、線が掠れて消えていくような表現が出来る。本研究の動画では、最も長い線の値を0.050、次に長い線と、短い線を0.030、最も短



い線を 0.010 に設定した。以上の処理を行うことで、境界線の動線が得られる。

ただし、モデルの移動スピードが速すぎると、一秒間に放出するパーティクルの量が間に合わず、線ではなく点に見えてしまうことがある。その場合は、パーティクルの Rate の値を上げる必要がある。また、カメラが近くなる場合、パーティクルの点が見えてしまうので、ポイントサイズを上げたり、Rate の値を上げたりすることで、近いカメラでも線に見せることが出来る。以下の図 3.15 にその様子を示す。

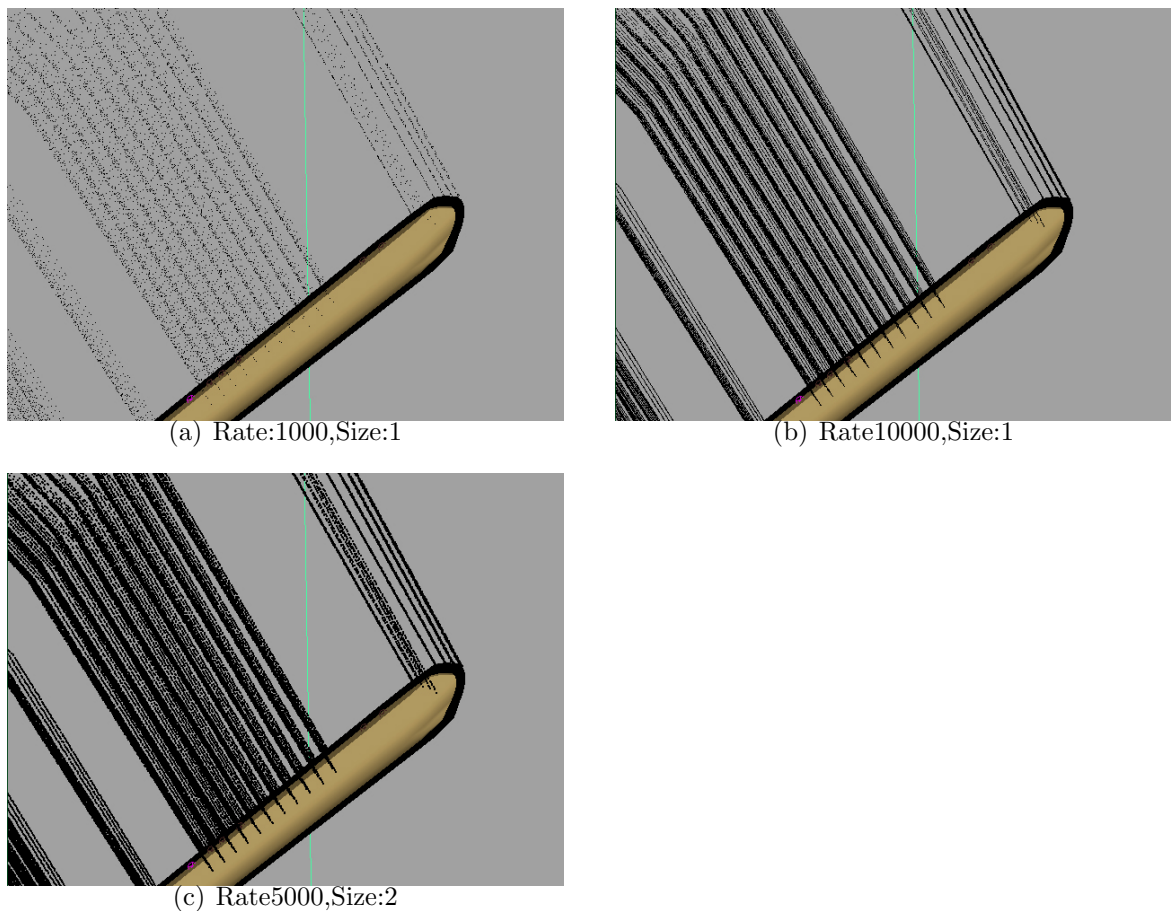


図 3.15: Rate の値とポイントサイズによる線の修正

図 3.15(a) は rate の値が 1000、ポイントサイズが 1 である。このように刀の部分にズームアップすると、パーティクルの点が見えてしまう。そこで、Rate の値

を 10000 に上げて修正したものが図 3.15(b) である。ズームになっていても線に見えることがわかる。ただし Rate の値は上げすぎると重くなってしまっているので調整が必要である。図 3.15(c) は Rate の値を 5000、ポイントサイズを 2 に設定した画像である。このようにポイントサイズも合わせて調整すると、Rate の値をそれほど高くしなくても線に見せることができる。

今回の動画では、図 2.1 で示したような表現にできるだけ近づけるため、図 2.1 のように前面は動線表現になっていない、という部分も再現する。それを再現するために、以下のような方法を用いた。境界線ブラーを適応する部分の境界線のモデルの複製モデルを作成し、レイヤーを分ける。複製モデルにおいて、移動の前面部分以外を消去する。こうすることで、複製モデルのレイヤーには境界線の前面だけが残る。以下の図 3.16 にその画像を示す。

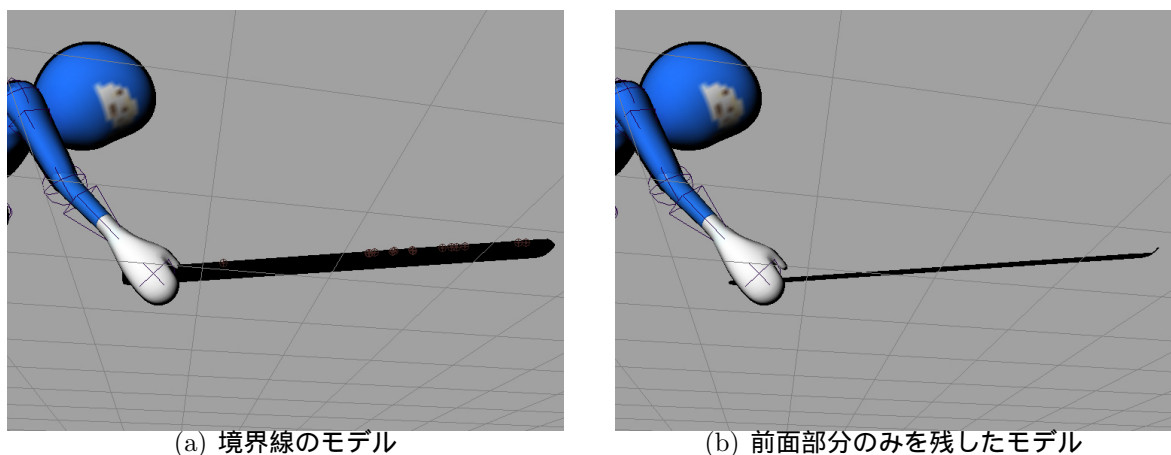


図 3.16: 境界線の前面の表現

図 3.16(a) に示されているような境界線のモデルの前面部分のみを残したのが、図 3.16(b) である。こうすることで移動の前面の境界線を表現することが出来る。

次に、境界の曖昧さを表現するために、Maya の標準機能で、レンダリング時にモデルにモーションブラーをかける。使用したのは 3D モーションブラーで、ブラーをかける強さの値 (Blur by Frame) は 2 に設定した。

以上の処理を行い、レンダリングを行う。本手法の場合、レンダertypeが points なので、パーティクルは Maya Hardware でしかレンダリングできない。しかし、トゥーンレンダリングに使用しているランプシェーダは Maya Hardware ではレンダリングができない。そこで動線は Maya Hardware でレンダリングし、モデルは Maya Software でレンダリングする。以下の図 3.17 にレンダリングした画像と、合成した画像を示す。

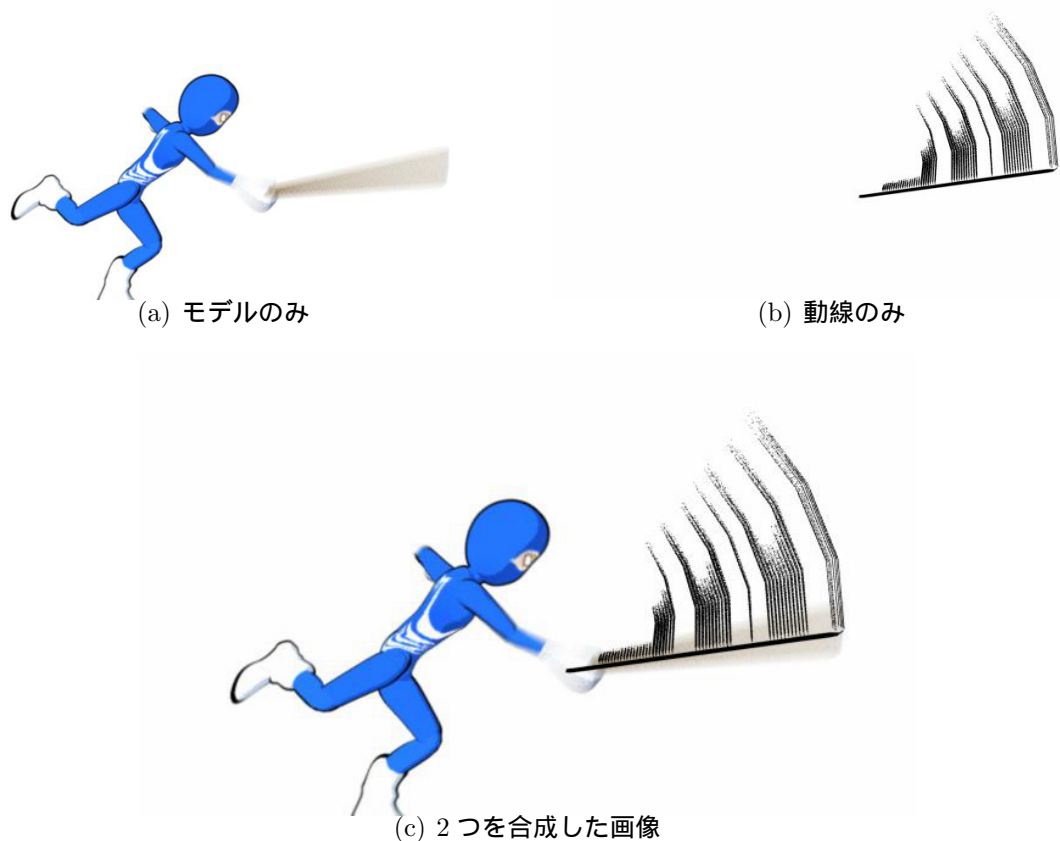


図 3.17: モデル画像と動線画像をそれぞれレンダリングし合成

図 3.17(a) がモデルをレンダリングした画像、図 3.17(b) が動線をレンダリングした画像である。これらの画像を After Effects で合成したものが、図 3.17(c) である。モデルの画像レイヤーを下に、動線の画像レイヤーを上を設定し、動線のレイヤーは乗算で合成する。この際に動線のレイヤーの不透明度を、最初は 0 に設

定し、動線が現れるフレームに近づくと徐々に値を上げ、動線が消えるときに徐々に下げる設定にすると、動線が滑らかに現れて消える、という動画になる。以上のような処理を行い、境界線ブラーの動画を制作した。

# 第 4 章

## 検証と考察

本章では本手法の検証と考察について述べる。4.1 節では、境界線のモデルを変形させる方法、パーティクルを使用する方法の2つの方法を用いて制作した境界線ブラーの動画が、漫画的なスピード誇張表現として3DCG上でも機能しているかについて、それぞれを検証した結果を述べる。4.2 節では、本手法を用いて境界線ブラーを描画する方法と、従来の手描きによって境界線ブラーを描く方法の制作時間を比較し、本手法が効率的かどうかの検証結果について述べる。4.3 節では、それぞれの方法の問題点の考察について述べる。

## 4.1 漫画的なスピード誇張表現として効果的かどうかの検証

### 4.1.1 境界線のモデルを変形する方法について

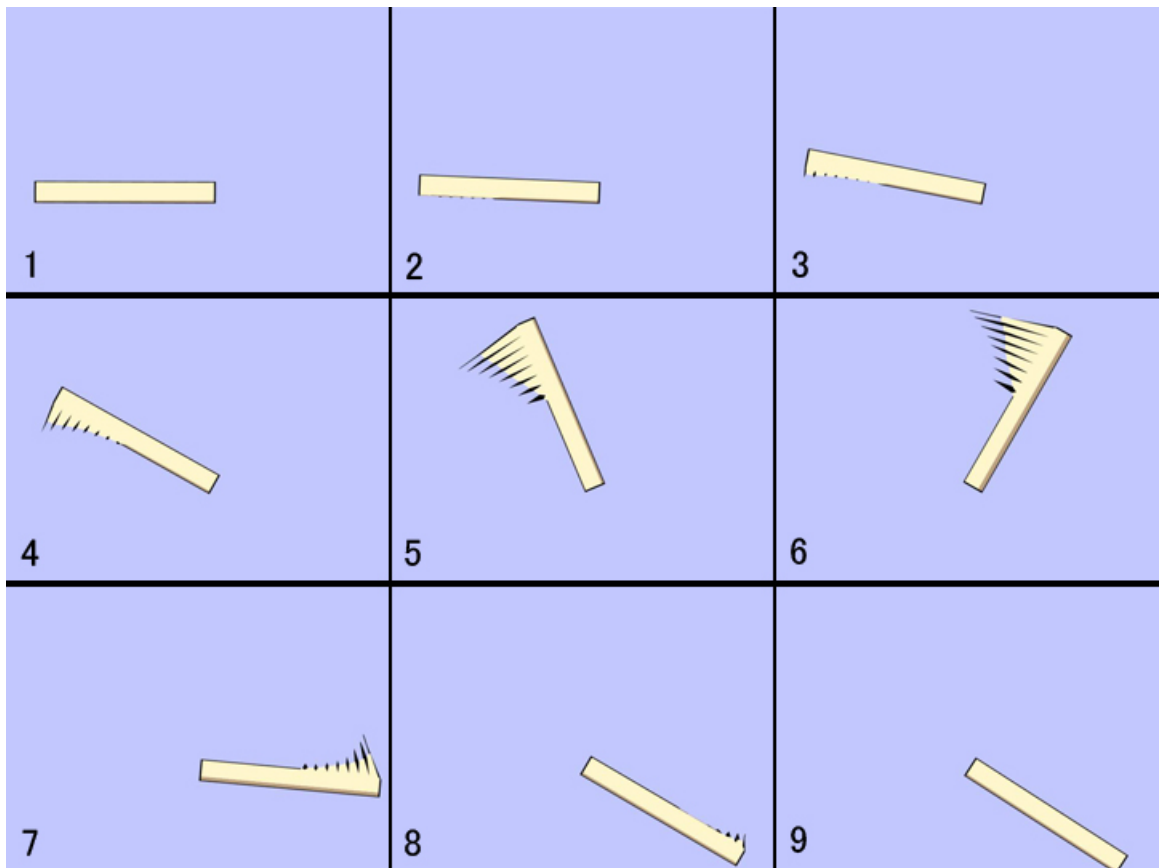


図 4.1: 実際に制作した動画の連番画像

図 4.1 が、境界線のモデルを変形する方法を用いて制作した動画の連番画像である。この動画を検証した結果を述べる。2.1 節で挙げた境界線ブラーの効果は、以下の 4 点であった。

- 境界の曖昧さによるスピードの誇張
- 線による移動方向の強調
- 線の長さによる移動幅の強調

- 線のランダム性を強調したスピードの誇張

この4点のうち、動線による移動の方向性の強調、線の長さによる移動幅の強調の2つは、この動画で効果が得られたといえる。線のランダム性を強調したスピードの誇張については、この動画では効果が得られなかった。理由として、動線の長い部分と短い部分をはっきりと分けた表現が境界線ブラーの特徴であるが、この動画においては、動線の本数が少なく、長い部分と短い部分の線の集まりが認識できない。よって、従来の効果線と大きく変わらない表現に見える。以下の図4.2にその違いを示す。

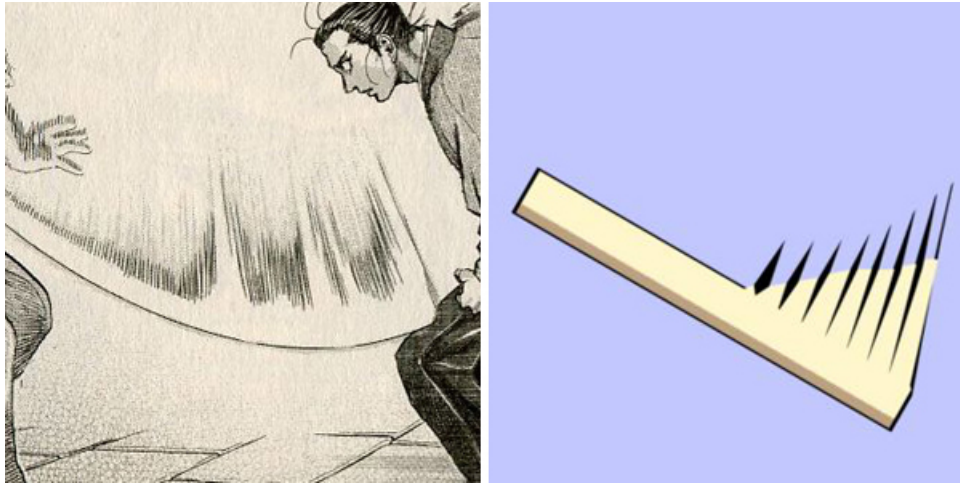


図4.2: 線のランダム性の強調についての違い(左図:山口貴由,南條範夫,秋田書店)

境界の曖昧さによるスピード誇張については、モデルの色と同じ色の板ポリゴンを用意することで境界の曖昧さを表現したが、その板ポリゴンと背景の境目ははっきりとしているため、完全に再現できたとは言えない。

また、この方法の問題点として、カメラの視点を変えてしまうと、引き伸ばした境界線のモデルが、線に見えなくなってしまうという問題がある。以下の図4.3にその状態を示す。



図 4.3: 別視点から見た時の線の破綻

このように、カメラの角度を変えてしまうと、まったく意図しない画像になってしまう。この方法では、カメラの位置を最初に決めて、境界線のアニメーションを作る必要がある。また、この方法では動線を曲線で表現したい場合、曲線を表現しにくいという問題がある。

結果として、今回の境界線のモデルを変形させる方法を使用して制作した動画に関しては、移動方向の強調、移動幅の強調の効果は得られたが、境界線ブラーの特徴である境界の曖昧さ、線のランダム性の強調については完全に再現できなかった。



#### 4.1.2 パーティクルを使用する方法について

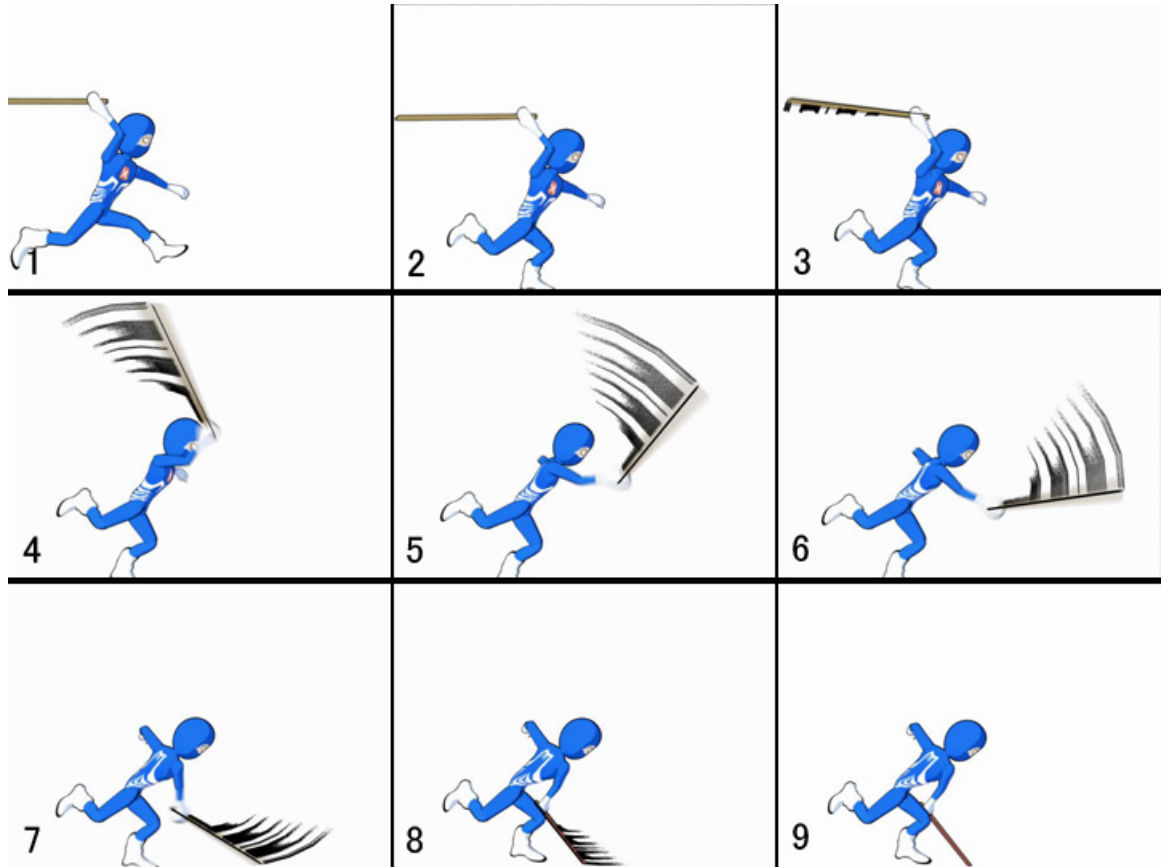
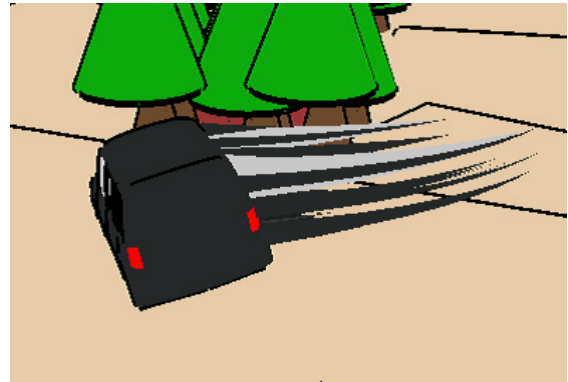


図 4.4: パーティクルを用いた境界線ブラーの動画の連番画像

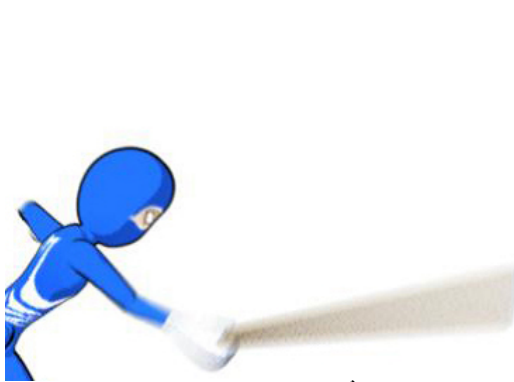
図 4.4 が、パーティクルを使用する方法を用いて制作した動画の連番画像である。この動画を検証した結果について述べる。結果として、この動画は漫画における境界線のブレ表現と同様の効果が得られたことがわかった。ただし、境界線ブラーの動画において、キャラクターの動きが速い場合、境界線ブラーによってブレている部分の詳細な変化がわからないため、従来の効果線表現と大きく変わらないことがわかった。しかしスローモーションなどキャラクターの動きが遅い演出の場合は、漫画における境界線のブレ表現と同様の効果を発揮した。また、従来手法と比較した結果、境界の曖昧さと線のランダム性の強調という効果において、境界線ブラーに有効性があるという結果を得た。以下に比較した図 4.5 を示す。



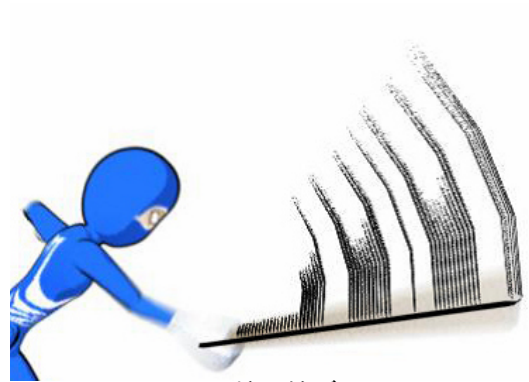
(a) 効果線



(b) ゆがみ



(c) モーションブラー



(d) 境界線ブラー

図 4.5: 効果を比較した画像

図 4.5(a)、図 4.5(b) は先行研究であるカートゥーンブラー [6] からの画像であり、それぞれが効果線、ゆがみ (アニメにおけるオバケ表現) を表現している。図 4.5(c) は従来のモーションブラー、図 4.5(d) が境界線ブラーの画像である。効果線、ゆがみの効果と境界線ブラーの比較の結果、本手法の特徴である境界の曖昧さ、線のランダム性の強調という点においては、本手法のほうが有用性があるといえる。従来のモーションブラーと境界線ブラーの比較の結果、線のランダム性の点で、本手法のほうが有用性があるといえる。

以上のことから、境界線ブラーは、スローモーションなどキャラクターの動きが遅い演出の場合、境界の曖昧さによるスピード誇張と移動の方向性を示す効果が期待できる、ということが言える。

ただし本手法の問題点として、以下のことが挙げられる。

- 乗算で動線レイヤーを重ねるので、陰面部分に隠れるはずの動線も表示されてしまう
- Maya のブラー機能が移動の後方だけにブラーを描画できないので、モデルのブレが移動の前面にも表示されてしまう
- パーティクルの点を大量に描画するため重くなってしまう

以下の図 4.6 にその問題の点を示す。

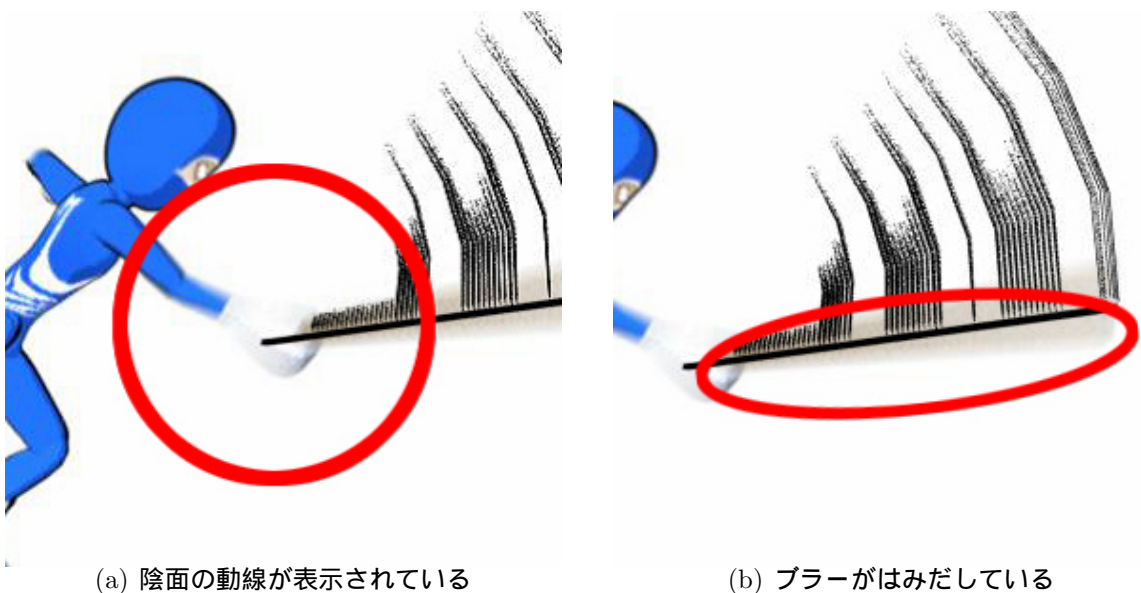


図 4.6: パーティクルを使用する方法での問題点

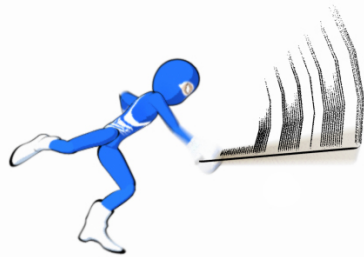
図 4.6(a) は手に隠れているはずの動線が表示されてしまっている問題、図 4.6(b) は前面の動線の前方にモデルのブラーがはみだしている問題を表している。この問題は、境界線を変形させる方法では解決される問題である。

## 4.2 効率化に関する検証

ここでは、境界線ブラーの動画を制作する際、従来のように手描きで制作した場合よりも本手法のほうが効率的であるかどうかについて検証した。具体的には、フォトショップで動線を描いて画像を1枚1枚生成していくのと、本研究の手法を使用した場合の作業時間を比較した。手描きの方法は、Mayaでレンダリングしたモデルの画像に、フォトショップのパスを用いて動線を描いていく。その方法と、本手法のパーティクルを用いて動線を描く方法で、どれだけ時間がかかるかを調査した。本手法のレンダリングに対する計測は、以下のようなスペックのPCにて行った。

- OS:Microsoft Windows XP Professional SP2
- CPU:Pentium4 CPU 3.20GHz
- メモリ:1.00GB RAM
- グラフィックボード:RADEON 9800 PRO

手描きの絵は1枚だけ描き、1枚にかかる時間×フレーム数で時間を算出した。この動画では動線が描画されるのは35フレームなので、35フレームで時間を計算した。以下の図4.7はそれぞれの方法で描いた画像である。



(a) レンダリング画像



(b) 手描き画像

図 4.7: 本手法のレンダリング画像と手描きによる画像

図 4.7(a) が本手法のレンダリング画像、図 4.7(b) が手描きによる画像である。この画像を制作した結果、以下のような結果を得た。パスによる手描きで 1 枚の画像を制作するのにかかった時間は 5 分 13 秒で、35 フレーム制作する場合は計算上 182.6 分かかる、という結果となった。次に本手法のレンダリング時間を計測した結果、35 フレームの動線レイヤーを書き出すのにかかる時間は、4 分 25 秒であった。よって、手描きによって動線を描くよりも、本手法を使用したほうが、効率的になったといえる。

### 4.3 問題点の考察

ここでは本手法の現状での問題点についての考察を述べる。まず境界線モデルの変形による方法について述べる。今回の動画制作では、境界線ブラーの特徴である効果を再現することができなかったが、動線のランダム性の強調と境界の曖昧さが表現しきれなかった問題については、表現方法を工夫することでより漫画の表現に近づけることができると考えられる。動線のランダム性の強調については、境界線のモデルの頂点数を増やし、表示できる動線の数を増やす。そしてアニメーションをつける際に動線の長さを調整すれば、制作者の意図にあわせてラ

ンダム性も表現できると考えられる。境界の曖昧さについては、モデルにモーショ  
ンブラーをの効果をかけることによって、曖昧さを表現することができる。しか  
し、その場合境界線にもブラーがかかってしまうので、境界線だけブラーをかけ  
ない処理が出来れば、より漫画の表現に近いものが制作できると考えられる。曲  
線が表現できない問題については、アニメーションの手間がかかるが、モデルの  
頂点数を増やし、曲線のようにアニメーションをつけることで、ある程度は表現  
できると考えられる。ただし、視点の変更に対応できない問題については、本研  
究では解決法は発見できなかった。境界線のモデルを変形させる手法の課題であ  
るといえる。

次に、パーティクルを使用する方法の問題点についての考察を述べる。移動の  
前面の動線よりもモデルのブラー部分がはみ出してしまう問題については、Maya  
上で移動の後方にしかブラーを描画させないプラグイン等を制作することで、こ  
の問題は解決できると考えられる。陰面が表示されてしまう問題、処理が重い問  
題については、本研究では解決法は発見できなかった。パーティクルを使用する  
方法の課題であるといえる。今後は以上のことを修正していけば、より3DCGに  
おける境界線ブラーの表現が効果を成すと考えられる。

## 第 5 章

### まとめ

本研究において、境界線のモデルを変形させる方法とパーティクルを使用する方法を用いて、境界線のブレによるスピード誇張表現を 3DCG 上で実現した。その効果を検証した結果、キャラクターの移動の速度が速い場合、その効果は従来の効果線による表現と大きな違いは無かった。しかしスローモーションなど、あえて動きを遅くする演出の場合に、この表現は動線のランダム性の強調によるスピード誇張の効果と、キャラクターの境界線の形が曖昧に表現されることによるスピード誇張という効果を得た。本研究の結果としては、パーティクルを使用する方法のほうが、境界線のモデルを変形させる方法より、漫画の境界線のブレ表現を再現できた。境界線のモデルを変形させる方法は、視点の変更に対応できない、境界の曖昧さが表現しきれていない、動線の曲線表現に手間がかかる、という問題がある。しかしパーティクルを使用する方法についても、陰面に隠れるはずの動線も表示されてしまう、モデルのブレが移動の前面にも表示されてしまい動線の前にはみ出す、パーティクルを大量に描画するため処理が重い、という問題があり、これらの点では境界線のモデルを変形させる方法のほうが優っているといえる。これらの問題は、今後の課題として解決すべき点である。

# 謝辞

最後に、この論文を書くにあたり、ご指導していただいた渡辺大地講師、ならびに様々なアドバイスをしていただいた講師の方々、研究についての相談を受けてくれた院生の方々、共に励ましあい研究に協力してくれた学部生の皆様に厚く御礼申し上げます。



## 参考文献

- [1] Amy Gooch, Bruce Gooch, Peter Shirley, Elaine Cohen, “A Non-Photorealistic Lighting Model For Automatic Technical Illustration”, SIGGRAPH 98, 1998.
- [2] Allison W. Klein, Wilmot Li Michael, M. Kazhdan Wagner, T. Correa, Adam Finkelstein, Thomas A. Funkhouser, “Non-Photorealistic Virtual Environments”, SIGGRAPH2000, 2000.
- [3] Emil Praun, Hugues Hoppe, Matthew Webb, Adam Finkelstein, “Real-Time Hatching”, SIGGRAPH2001”, 2001.
- [4] 丹治宏文, “リアルタイム 3DCG ツールキット上での絵画調レンダリングの実装とその効果に関する研究”, 東京工科大学大学院メディア学研究科卒業論文, 2002.
- [5] 菅野博之, 「快描教室」, 美術出版社, 1997.
- [6] 大林正一, 近藤邦雄, 今間俊博, 岩本賢一 “3DCG アニメーションのためのカトゥーンブレンダー”, IGDA 日本 SIG-GT 第 8 回「ノンフォトリアリスティックにおける生産性の向上へのアプローチ」, 2006
- [7] 中村陽介 “3DCG アニメーションにおける効果線表現手法の提案と効率化に関する研究”, 東京工科大学大学院メディア学研究科修士論文, 2004.

- [8] Maic Masuch, Stefan Schlechtweg, Ronny Schulz, “Speedlines Depicting Motion in Motionless Pictures”, SIGGRAPH99, 1999.
- [9] Comickers テクニックブック, 「マンガスタートアップガイド ペン&インク」, 美術出版社, 2000.
- [10] 山口貴由, 南條範夫, 「シグルイ」, 秋田書店, 第3巻, 2005.
- [11] マンガひょうげんろんサイト, マンガびと,  
<<http://www.space.lan.ne.jp/~taro/index.html>>.
- [12] Enterbrain, デジタルファミ通ホームページ,  
<<http://www.enterbrain.co.jp/digifami/index.html>>.
- [13] えにくま, 役に立たないアニメーション用語集,  
<<http://www4.ocn.ne.jp/~kagi/index.html>>.
- [14] 川瀬正樹, MasakiKawase'sHomePage,  
<<http://www.daionet.gr.jp/~masa/index.html>>.
- [15] shred of game, <<http://platz.jp/~moal/game.html>>.
- [16] 松浦健一郎, 司ゆき, 「ゲームエフェクトマニアックス」, ソフトバンククリエイティブ, 2006.
- [17] t-pot, PROGRAMMING,  
<<http://tpot.jpn.ph/t-pot/program/index.html>>.
- [18] 杉谷泰宏, 「Maya3D スーパーテクニク」, ソーテック社, 2003.
- [19] 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.