

2011年度 卒業論文

3DCGを用いた背景制作における  
手描きアニメ調ブラーの表現

指導教員：渡辺 大地 講師  
三上 浩司 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 M0108492  
渡辺 那

**2011年度 卒業論文概要**

**論文題目**

3DCG を用いた背景制作における  
手描きアニメ調ブラーの表現

**メディア学部**

**学籍番号： M0108492**

**氏名**

渡辺 那

**指導  
教員**

渡辺 大地 講師  
三上 浩司 講師

**キーワード**

3DCG、誇張表現、モーションブラー、  
ノンフォトリアリスティック

近年、3DCGは技術力や効率性の向上によって、テレビ番組やゲーム等に幅広く利用されている。従来は手描きによる表現が主流であったアニメ作品の中にも、3DCGを利用したものが多く存在するようになった。それに伴い、手描きの質感に馴染むようにノンフォトリアリスティックな表現や演出の研究も盛んになり、漫画やアニメの特有の表現や誇張表現も増えてきた。この、アニメ特有の誇張表現の中に、高速で動く物体の輪郭を歪ませブラー表現がある。この輪郭を歪ませ手描きアニメ特有のブラー表現は、3DCGでよく使う写実的なブラーでは表現できない。3DCGで手描きアニメ特有のブラー表現をする場合は、輪郭が歪んで見えるように3Dモデルの頂点の位置を1つ1つ細かく移動させて、表現する手法がある。しかし、この手法では非常に手間がかかる。そこで3Dモデルの移動方向に対して、後方の輪郭を歪ませアニメ特有のブラーを表現した研究がある。しかし、この手法は3Dモデル自体の移動速度に依存して歪んでいる。そのため、背景のように大量の3Dモデルで表現しようとした場合には、また別の手法を考えなければならない。本研究では手描きアニメにみる背景を鋸歯形状に歪ませブラー表現(以下「ジャギーブラー」)を効率的に3DCGで実現することを研究対象とする。

本手法では、画面の中で物体がどのように動いて見えたかという、見かけ上の動きに対してジャギーブラーを生成する。カメラの移動方向からジャギーブラーを生成する輪郭を検出し、カメラと各3Dモデルの相対速度から、見かけ上の動きに対して、前方の輪郭か後方の輪郭かを求める。前方の輪郭には物体がへこんでいるようなジャギーブラーを作る。そして、後方の輪郭には物体が伸びているようなジャギーブラーを作る。ジャギーブラーは3Dモデルの輪郭に3角形ポリゴンを作る。そして、ポリゴンの色を前方は3Dモデルの進行方向にある色、後方は3Dモデルの色で塗るという方法でそれぞれ、へこんだり伸びたりして見えるような歪みを作る。

提案したアルゴリズムに従い、物体にジャギーブラーをつけるプログラムを実装し、表現の検証を行った。その結果、物体を平行して追うようなカメラの動きに対して、効率的にジャギーブラーを生成することができた。

# 目次

<b>第1章</b>	<b>はじめに</b>	<b>1</b>
1.1	研究の背景と目的 . . . . .	1
1.2	本論文の構成 . . . . .	4
<b>第2章</b>	<b>背景の手描きアニメ調ブラー</b>	<b>5</b>
2.1	背景の手描きアニメ調ブラーとは . . . . .	5
2.2	3DCGにおけるブラー表現 . . . . .	6
<b>第3章</b>	<b>提案手法</b>	<b>11</b>
3.1	スクリーン空間を用いたジャギーブラーの パラメーターの取得 . . . . .	12
3.1.1	輪郭を構成しているピクセルの検出 . . . . .	12
3.1.2	相対速度の検出 . . . . .	14
3.1.3	見かけ上の動きの前方と後方の検出 . . . . .	16
3.1.4	ジャギーブラーの色の指定 . . . . .	17
3.2	3次元空間でのジャギーポリゴンの生成 . . . . .	17
3.2.1	3次元空間の座標指定 . . . . .	17
3.2.2	ジャギーブラーの生成 . . . . .	18
<b>第4章</b>	<b>検証と考察</b>	<b>20</b>
4.1	検証 . . . . .	21
4.2	考察 . . . . .	22
<b>第5章</b>	<b>まとめ</b>	<b>24</b>
	<b>謝辞</b>	<b>25</b>
	<b>参考文献</b>	<b>26</b>

# 目 次

1.1	写実的なブラーの例 . . . . .	2
1.2	手描きアニメにおける動きの表現 . . . . .	3
2.1	ジャギーブラーの例 . . . . .	6
2.2	元の画像 . . . . .	7
2.3	写実的なブラーをかけた画像 . . . . .	7
2.4	「線」によるスピード表現 . . . . .	8
2.5	「残像」によるスピード表現 . . . . .	8
2.6	「ゆがみ」によるスピード表現 . . . . .	8
2.7	ラティスポイントの格子 . . . . .	10
2.8	ラティスポイントの移動の様子 . . . . .	10
3.1	スクリーンスペースの図解 . . . . .	11
3.2	スクリーンスペース上の映像 . . . . .	14
3.3	輪郭ピクセルの検出 . . . . .	14
3.4	相対速度を色情報に変換 . . . . .	16
3.5	ジャギーブラーの色の指定 . . . . .	17
3.6	ジャギーポリゴンの座標指定 . . . . .	18
3.7	ジャギーブラーの生成 . . . . .	19
4.1	本手法を用いたジャギーブラーの連番画像 . . . . .	21
4.2	相対速度に応じたジャギーブラーの描画 . . . . .	22
4.3	実際の動きと異なる方向にブラーがでている例 . . . . .	23

# 第 1 章

## はじめに

### 1.1 研究の背景と目的

近年、3次元コンピュータグラフィックス (以下、3DCG) の技術力や効率性が向上している。そのため、テレビ番組やゲーム等にも 3DCG が利用されている。従来は手描きによる表現が主流であった手描きアニメ作品の中にも、3DCG を利用したものが多く存在するようになった。その背景としてコンピュータの性能や 3DCG ソフトの性能が向上したことにより、従来の手描きアニメでは困難だった表現が可能になり、作品の映像表現力が格段に広がったことがある [1]。

また、3DCG の技術力が向上していき、さまざまなレンダリング手法が研究されている。その中で、写実的ではない、油絵調や鉛筆画調、セル画調などの人の手で描いたような画像を、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 [2][3][4] が盛んに行われている。ノンフォトリアリスティックな表現は質感やタッチの研究だけでなく、漫画で用いる特有の表現や手描きアニメで用いる誇張表現といった、動きや演出に関する研究 [5][6][7] も行われている。動きの表現に関して手描きアニメでは、高速に動く物体に特有な表現を用いることがある。

通常の 3DCG で使う高速に動く物体の表現は、実際のカメラで撮影した映像を参考にしてしている。実際のカメラでは、高速で移動する物体を撮影すると、物体がブレて写る。これは、撮影する際のシャッタースピードが遅い事が原因で起こる。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

いた分だけ伸びたように撮影される。そのため、物体の動く移動速度が速かったり、写す瞬間にカメラを素早く動かして、位置を大きく変えたりするとブレ幅が大きくなる。CG アニメでは映像を作る場合にシャッタースピードという概念がないので、物体がブレて映ることはない。しかし、写実的な動きを表現するために意図的にブレの表現を加える事がある。この手法をモーションブラーと呼ぶ [8]。図 1.1 に写実的なブラーの例を示す。



図 1.1: 写実的なブラーの例

しかし、手描きアニメで使う高速に動く物体の表現は、写実的なブラーでは表現できない。手描きアニメでの表現はブレで表現するのではなく、のこぎりの歯のような鋸歯形状に輪郭を歪ませるといった特有の表現をすることがある。この歪みは、実際のカメラで撮影するとき、物体の移動速度が速いとブレが大きくなるように、移動速度が速いほど歪みが大きくなる。高速に動いて見えるときに物体の輪郭部分にできる鋸歯形状のことを「ジャギーブラー」と呼称する。図 1.2 にジャギーブラーを使ったスピード表現を示す。赤い丸で囲んでいる部分がジャギーブラーのでている部分である。

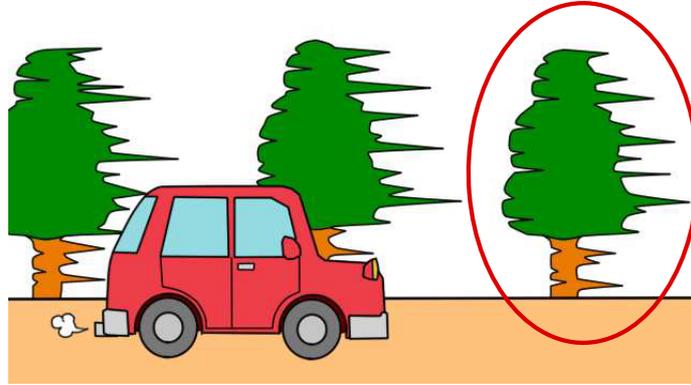


図 1.2: 手描きアニメにおける動きの表現

写実的なブラー表現では、ジャギーブラーの鋸歯形状に歪む表現ができない。そのため、3DCG 上で手描きアニメ調のジャギーブラーを表現する場合は、物体の移動に合わせて物体の輪郭を細かく鋸歯形状に歪ませる処理をする。だが、3DCG で鋸歯形状に歪ませるには、物体の移動方向や移動速度を考えながら、頂点 1 つ 1 つの位置を動かすことになるので、大変である。また、複数の 3D モデルにジャギーブラーをつけようとした場合は、モデルの数だけ頂点を動かす手間が増える。

このため本研究では、手描きアニメにみる背景へのジャギーブラーに着目し、カメラの移動に対して方向や速度を考慮した歪みがでるアルゴリズムを提案する。このアルゴリズムを用いて、手描きアニメにみる背景へのジャギーブラーを効率的に 3DCG 上で実現することを目的とした。提案手法では、画面の中で物体がどのように動いて見えたかという、見かけ上の動きに対してジャギーブラーを生成する。カメラの移動方向からジャギーブラーを生成する輪郭を検出し、カメラと各 3D モデルの相対速度から見かけ上の動きに対して、前方の輪郭か後方の輪郭かを決める。前方の輪郭には物体がへこんでいるようなジャギーブラーを作り、後方の輪郭には物体が伸びているようなジャギーブラーを作る。ジャギーブラーは輪郭に 3 角形ポリゴンを作る。そして、ポリゴンの色については前方の輪郭は 3D モデ

ルの進行方向にある色で、後方の輪郭は3Dモデルの輪郭の色で塗るという方法で歪んでいるように見せる。提案したアルゴリズムに従い、物体にジャギーブラーをつけるプログラムを実装し、表現の検証を行った。その結果、物体を平行して追うようなカメラの動きに対して、ジャギーブラーを生成することができた。

## 1.2 本論文の構成

本論文は全5章で構成する。第2章で手描きアニメ調ブラーについて述べ第3章では提案手法を述べる。第4章では実装結果の検証と考察について述べる。最後に、第5章では本研究のまとめと、今後の展望を述べる。

## 第 2 章

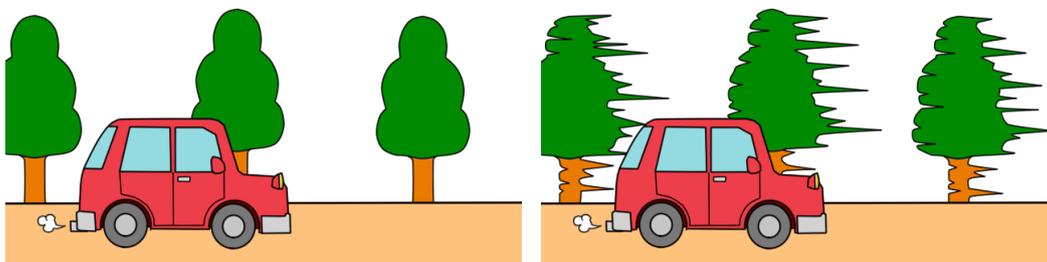
# 背景の手描きアニメ調ブラー

本研究では背景の手描きアニメ調のブラーに注目した。2.1 節では、手描きアニメにおける背景のブラー表現について述べる。2.2 節では、3DCG におけるブラー表現について述べる。

### 2.1 背景の手描きアニメ調ブラーとは

手描きアニメにおける動きの表現には、物体が高速に移動する際、物体の後方の輪郭を鋸歯形状に変形する表現がある [9][10]。また、作品によっては、前方の輪郭もへこむように歪ませる場合もある。

手描きアニメではキャラクターの表情や動きを見せるために、車やバイクを画面の中央に捉え、背景の方を動かすことがある。このとき、背景をただ動かしているだけではスピード感がなくなるので、背景の形を歪ませたりする [11]。図 2.1(a)、図 2.1(b) は手描きアニメでの背景を動かす演出の例である。



(a) ジャギーブラーがでていない背景

(b) ジャギーブラーがでている背景

図 2.1: ジャギーブラーの例

図 2.1(a) がジャギーブラーがでていない背景であり、図 2.1(b) がジャギーブラーがでている背景である。この図では、車は右方向に向かって動いており、カメラの動きも車が常に画面の中に映るように車を追って右方向に動いている。この場合、木などの背景は実際には動いていないが、画面を写しているカメラが動いているので相対的にカメラの移動方向とは逆の左方向に向かって動いて見える。背景は見かけの上の移動方向に対して、前方である左側の輪郭がへこむように歪み、後方である右側の輪郭が伸びるように大きく歪む。

背景に使うブラー表現は背景そのものが移動するのではなく、カメラを素早く動かしたことにより、写している背景が流れて見える時に利用する [12]。この時のブラー表現はカメラの移動速度を使う。カメラの移動速度は、過去のフレームのカメラの位置と現在のフレームのカメラの位置の差分から求める。

## 2.2 3DCG におけるブラー表現

3DCG におけるモーションブラーには、写実的な表現と非写実的な表現の 2 つがある。

写実的な表現は、過去のフレームと現在のフレームの間の画像を任意の数描画

し、描画した画像を平均化する方法 [13] や、過去のフレームの画像を薄く描画して残像の効果を出す方法 [14] がある。図 2.2、図 2.3 は過去のフレームの画像を薄く重ねて残像効果を出した写実的なブラーの例である。

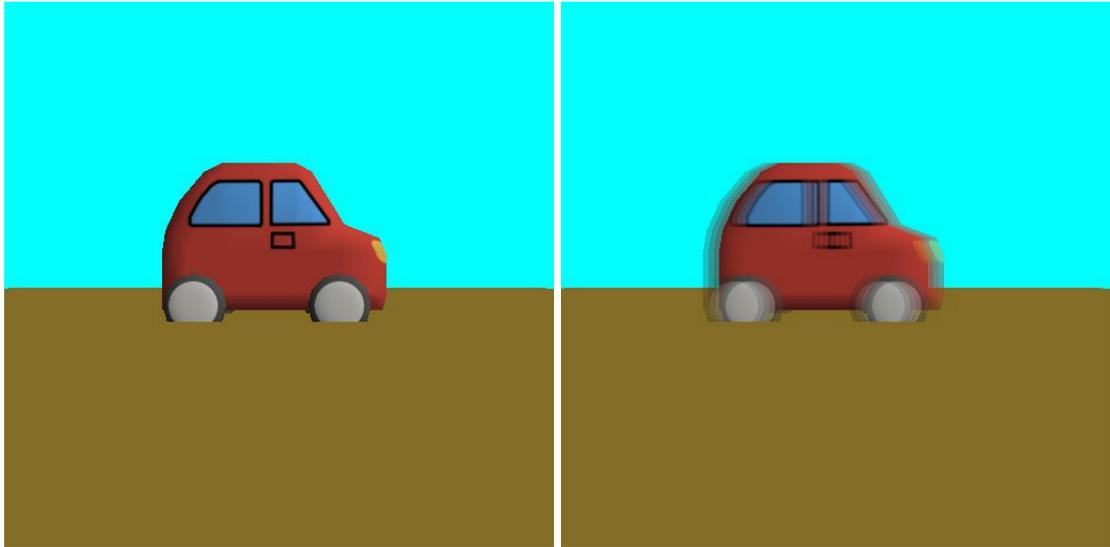


図 2.2: 元の画像

図 2.3: 写実的なブラーをかけた画像

これらの写実的な表現は、実際のカメラで撮影したようなブレの表現はできる。過去の画像を描画する方法では、輪郭を鋸歯形状に歪ませる表現ができない。

非写実的な表現は、大林らの研究 [15][16][17] がある。この研究では、手描きアニメのブラー表現を「線」、「残像」、「ゆがみ」の3種類に分類し、3DCG 上で表現している。次の図 2.4、図 2.5、図 2.6 に「線」、「残像」、「ゆがみ」をそれぞれ実装した画像を示す。



図 2.4: 「線」によるスピード表現



図 2.5: 「残像」によるスピード表現



図 2.6: 「ゆがみ」によるスピード表現

この研究での「ゆがみ」の表現は、動いているオブジェクトの移動方向に対して反対方向を計算する。そして、面を細かく分割したり、いくつかの面を結合したりすることで歪みの細かさを決める。ゆがみの細かさを決めた後に、移動方向の後方の面に対して、移動の軌跡に沿ったポリゴンを生成することで歪んでいるように見せている。

この研究によって、3DCG 上での後方の輪郭のジャギーブラーが実現されている。しかし、この手法は新たにポリゴンを生成しているため、前方の輪郭をへこむように歪ませることができない。また、3D モデル自体の移動速度に依存して歪ませているため、画面内全ての 3D モデルに対しカメラの動きとの相対的な動きから歪ませることができない。

その他に、ジャギーブラーを表現しようとする場合、3D モデルそのものを変形させる手法がある。この手法ならば、前方の輪郭をへこませ、後方の輪郭を伸びるように歪ませることができる。しかし、3D モデルの形状変形を用いてジャギーブラーを制作すると、カメラの動きに合わせて 3D モデルの頂点位置を細かく調整する必要がある。カメラの動きを意識しながら、頂点位置を決めることは手間がかかる。

また、3D モデルの変形手法の一種で、ラティス変形というものがある。これは 3D モデルの頂点を直接移動させるのではなく、3D モデル変形用の格子を当てはめ、その格子の頂点を移動することで 3D モデル全体を変形する手法である。だが、頂点位置を細かく調整する必要があるという問題点は変わらない。次の図にラティスによる変形の例を示す。図 2.7 は、3DCG モデルにラティスポイントの格子を割り当てた図である。図 2.8 は、ラティスポイントの移動により 3DCG モデルを変形させている図である。

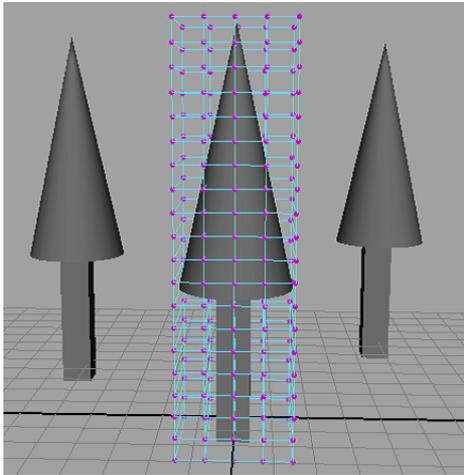


図 2.7: ラティスポイントの格子

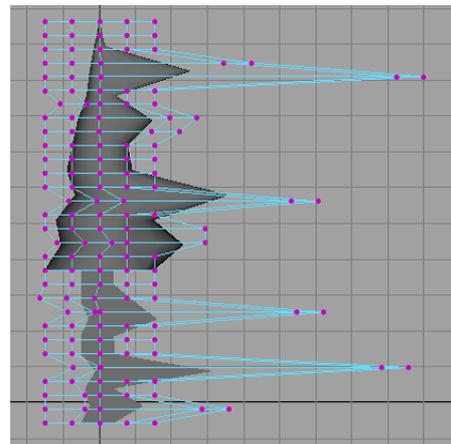


図 2.8: ラティスポイントの移動の様子

3D モデルの頂点位置を変えてジャギーブラーを生成する手法は 3D モデルが増えていくごとに手間が増えていく。従って、背景のように多くのポリゴンを使う場合にはこれらの手法を使ってジャギーブラーを作るのは困難である。この問題を解決するために 3D モデルを描画する画面の情報（スクリーンスペース情報）を使ってジャギーブラーを生成する手法を提案する。

# 第 3 章

## 提案手法

本章では、3DCG でジャギーブラーを表現する手法を述べる。本手法ではスクリーンスペースの情報からブラーをかける部分を検出し、3 角形ポリゴンを生成することでジャギーブラーを描画する手法である。スクリーンスペースとは、カメラが 3D モデルを描画する画面の領域のことである。図 3.1 にスクリーンスペースの図解を示す。

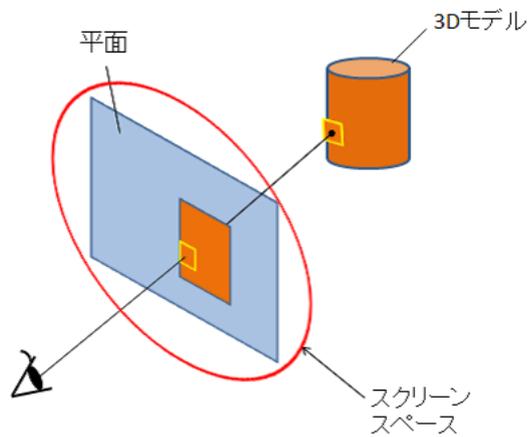


図 3.1: スクリーンスペースの図解

ジャギーブラーは前方と後方の輪郭でジャギーの出方が異なる。そのため、ジャギーブラーの実現には、輪郭の検出をし、見かけ上の移動方向の前方と後方を判

断する。輪郭の歪みは速度が速いほど大きく歪むため、物体とカメラの相対速度が大きいほど歪みを大きくする。ジャギーブラーを画面上で表現するために輪郭の位置に3角形ポリゴンを生成する。ジャギーブラーが発生する際、前方の輪郭はへこむように小さく歪む。そのため、歪ませる3Dモデルの見かけ上の進行方向にある色でジャギーブラーを生成し、へこんでいるような歪みを表現する。また、後方の輪郭は大きく伸びるように歪むので、後方の輪郭の色でジャギーブラーを生成する。

3.1節では、スクリーンスペースを用いたジャギーブラーのパラメーターの取得をする手法について述べる。3.2節では、3次元空間でのジャギーポリゴンの生成する手法について述べる。

## 3.1 スクリーンスペースを用いたジャギーブラーのパラメーターの取得

### 3.1.1 輪郭を構成しているピクセルの検出

ジャギーブラーはカメラが横方向に動いた場合は、3Dモデルの左右の輪郭に、縦方向に動いた場合は、上下の輪郭にでる。ジャギーブラーを生成するために、ジャギーブラーのでる輪郭をスクリーンスペース上で検出する。

ジャギーブラーのでる輪郭を検出するためにデプスバッファを利用する手法[18]を用いる。デプスバッファとは、スクリーン上の各ピクセルに描画したものが視点からどれだけ離れた位置に存在するかという深度値をメモリ領域に格納する技術のことである。3Dモデルの輪郭となる部分は、境目となるピクセルの間で深度値に大きい差がでる。従って、スクリーンスペース上で隣接するピクセルの深度値を比較していき、物体の輪郭を構成しているピクセル(以下、輪郭ピクセルと呼称する)を検出する。

深度値を比較するピクセルはカメラの速度ベクトルによって変わる。カメラの速度ベクトル  $\mathbf{V}$  は、視点座標系における現在のカメラの位置  $\mathbf{P}$ 、過去のカメラ

の位置  $\mathbf{P}'$  とし、式 (3.1) で求める。

$$\mathbf{V} = \mathbf{P} - \mathbf{P}' \quad (3.1)$$

求めたベクトル  $\mathbf{V}$  から比較するピクセルが決まる。比較するピクセルによって、見かけ上の動きが上下左右のどの方向動いているかがわかる。比較するピクセルを決めるための実数  $i, j$  には、それぞれ式 (3.2)、式 (3.3) で求めた値を代入する。 $i$  は左右の動きを判断し、 $j$  は上下の動きを判断する。

$$i = \begin{cases} 1 & (\mathbf{V}_x > 0 \text{ または } \mathbf{V}_x < 0 \text{ のとき}) \\ 0 & (\mathbf{V}_x = 0 \text{ のとき}) \end{cases} \quad (3.2)$$

$$j = \begin{cases} 1 & (\mathbf{V}_y > 0 \text{ または } \mathbf{V}_y < 0 \text{ のとき}) \\ 0 & (\mathbf{V}_y = 0 \text{ のとき}) \end{cases} \quad (3.3)$$

各ピクセルを比較し、深度値の差が大きいところを輪郭の境目とし、比較したピクセルの内、深度値が小さいほうのピクセルを輪郭ピクセルとする。これは、深度値が小さいほうが視点に近い位置にある 3D モデルとなるからである。スクリーンスペースの座標を画面の左下を  $x = 0, y = 0$  とし、各ピクセルの深度値  $d$  とするとき、輪郭ピクセルの座標  $(p, q)$  は、式 (3.4) で求める。

$$(p, q) = \begin{cases} (x, y) & (d_{x,y} < d_{(x+i),(y+j)} \text{ のとき}) \\ (x + i, y + j) & (d_{x,y} > d_{(x+i),(y+j)} \text{ のとき}) \end{cases} \quad (3.4)$$

式 (3.1) を用い、物体の輪郭を検出したイメージ図を図 3.2、図 3.3 に示す。

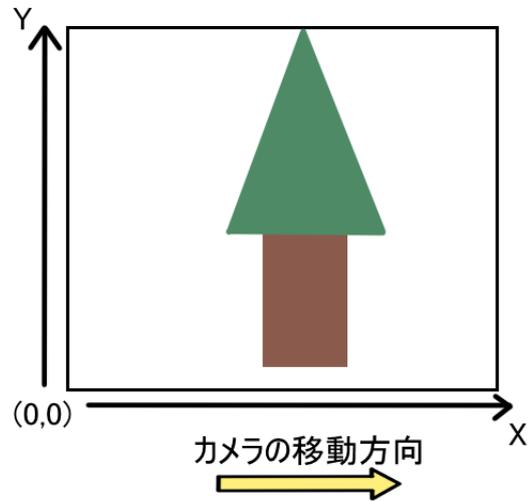


図 3.2: スクリーンスペース上の映像

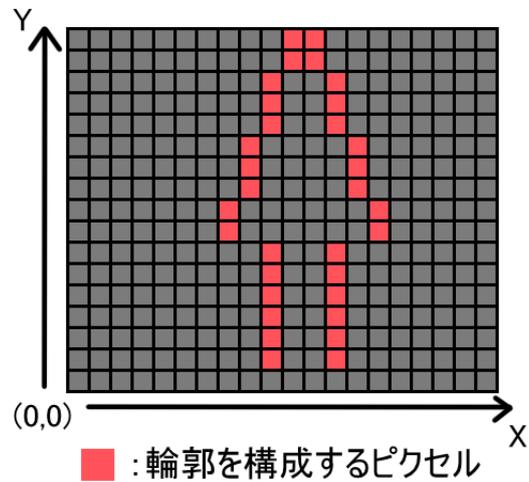


図 3.3: 輪郭ピクセルの検出

### 3.1.2 相対速度の検出

見かけ上の動きに応じたジャギーブラーをだすために、カメラと 3D モデルの相対速度を求める。3D モデルの速度ベクトル  $S$  は、視点座標系における現在の 3D

モデルの位置  $M$ 、過去の 3D モデルの位置  $M'$  とし、式 (3.5) で求める。

$$\mathbf{S} = \mathbf{M} - \mathbf{M}' \quad (3.5)$$

カメラと 3D モデルの相対速度  $\mathbf{B}$  は、式 (3.1) で求めたカメラの速度ベクトル  $\mathbf{V}$  と、式 (3.5) で求めた 3D モデルの速度ベクトル  $\mathbf{S}$  から、式 (3.6) で求める。

$$\mathbf{B} = \mathbf{S} - \mathbf{V} \quad (3.6)$$

相対速度  $\mathbf{B}$  をスクリーンスペース上で扱うために各ピクセルに格納する。各ピクセルに格納するために線形補間 [19] を使い、相対速度  $\mathbf{B}$  を色情報  $\mathbf{L}$  に変換する手法を用いた。 $\mathbf{B}$  の  $(x,y,z)$  をそれぞれ  $\mathbf{L}$  の  $(R,G,B)$  に変換する。この時、RGB のそれぞれの値の範囲を 0.0~1.0 とする。ジャギーブラーが最大まで伸びる相対速度を決める。スクリーンスペース上での正の方向の最大値を  $\mathbf{B}_{max}$ 、負の方向の最大値を  $\mathbf{B}_{min}$  とする。これ以上、相対速度が速くなってもジャギーブラーの長さは伸びない。色情報  $\mathbf{L}$  の  $(R,G,B)$  の値それぞれを次の式 (3.7) で求める。

$$\mathbf{L} = \frac{\mathbf{B} - \mathbf{B}_{min}}{\mathbf{B}_{max} - \mathbf{B}_{min}} \quad (3.7)$$

スクリーンスペース上の 3D モデルの色を  $\mathbf{L}$  で塗りつぶした画像を描画し、各ピクセルに相対速度の情報を格納する。スクリーンスペース上の 3D モデルの相対速度を色情報に変換したイメージ図を図 3.4 に示す。

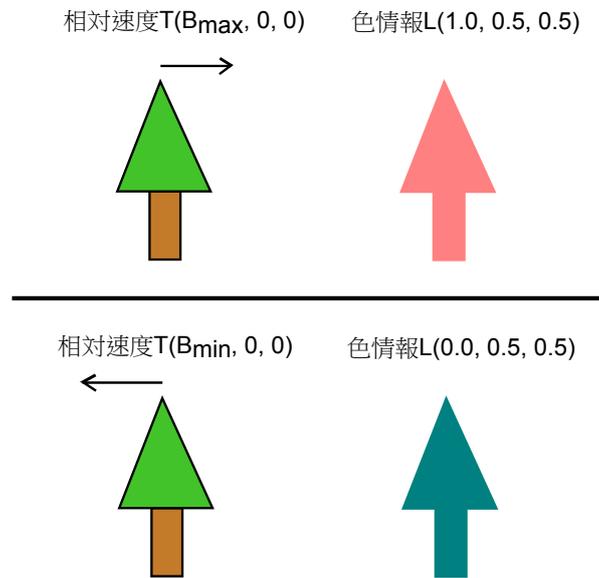


図 3.4: 相対速度を色情報に変換

### 3.1.3 見かけ上の動きの前方と後方の検出

3.1.1 節でジャギーブラーのでの輪郭を検出することはできた。しかし、物体の見かけ上の移動方向に対して、前方の輪郭と後方の輪郭でジャギーブラーの歪み方が異なる。異なるジャギーブラーを生成するため、輪郭ピクセルが前方の輪郭を構成するのか、後方の輪郭を構成するのかを区別する必要がある。スクリーンに表示した物体の見かけ上の移動方向に対して、前方の輪郭と後方の輪郭を区別する。

前方と後方の検出は相対速度と深度値を用いる。前方の輪郭は相対速度の進行方向側に輪郭の境目がある。後方の輪郭は進行方向と逆側に輪郭の境目がある。従って、輪郭ピクセルに対して、相対速度の進行方向側に隣接するピクセル、進行方向と逆側に隣接するピクセルの2つのピクセルと深度値を比較する。

深度値を比較し、進行方向側のピクセルの深度値が大きいときは前方の輪郭、逆側の深度値が大きいときは後方の輪郭と判断する。

### 3.1.4 ジャギーブラーの色の指定

ジャギーブラーの色は前方と後方で異なる。なぜなら、前方の輪郭は物体がへこんで見えるように歪み、後方の輪郭は物体が伸びるように大きく歪むからである。前方の輪郭は進行方向側に隣接したピクセルの色でジャギーブラーを生成し、前方の輪郭をへこんでいるように見せる。後方の輪郭は輪郭のピクセルの色でジャギーブラーを生成し、後方の輪郭を伸ばしているように見せる。ジャギーブラーの色の指定のイメージ図を図 3.5 に示す。

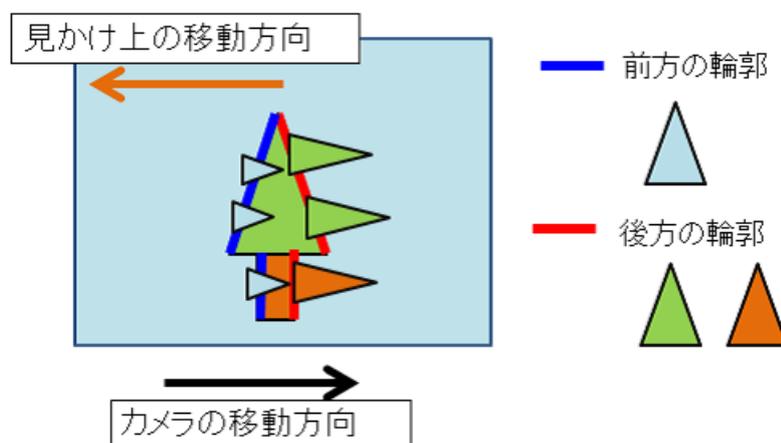


図 3.5: ジャギーブラーの色の指定

## 3.2 3次元空間でのジャギーポリゴンの生成

### 3.2.1 3次元空間の座標指定

3.1 節で前方の輪郭ピクセルと後方の輪郭ピクセルのスクリーンスペース上の座標を求めた。これを利用し、本節でジャギーブラーの3角形状を形成する。スクリーンスペース上で3角形の形に描画領域を指定し、ピクセルを塗りつぶしていく処理は負荷がかかる。そのため、3Dモデルの輪郭部分に接する座標に3角形状のポリゴン(以下、ジャギーポリゴンと呼称する)を配置する手法を用いた。

ジャギーポリゴンを配置するために、視点からスクリーンスペース上の輪郭ピクセルを通過して3Dモデルに接する点の座標を求める。その座標にスクリーンスペースと平行になるようにジャギーポリゴンを配置する。この時、前方の輪郭のジャギーポリゴンが3Dモデルに埋まってしまうことがあるため、ジャギーポリゴンは常に最前面に表示する。ジャギーポリゴンの座標指定のイメージ図を図3.6に示す。

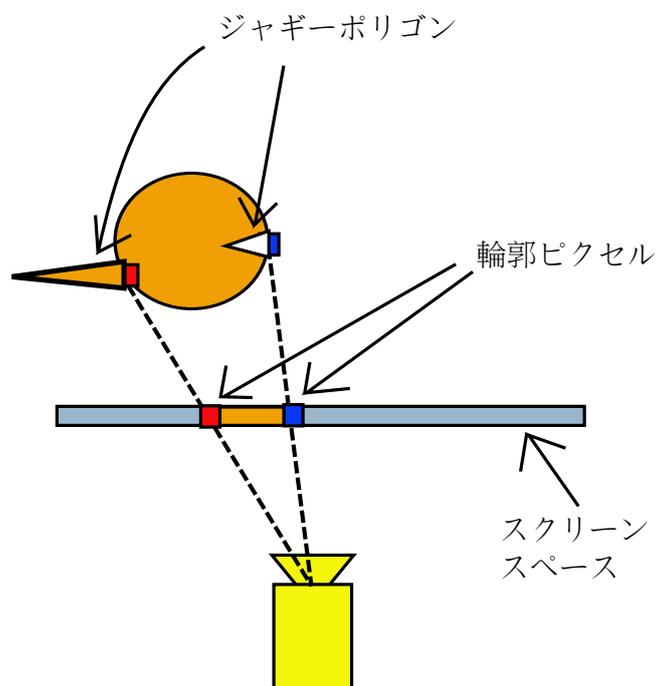


図 3.6: ジャギーポリゴンの座標指定

### 3.2.2 ジャギーブラーの生成

3.2.1節で求めた座標にジャギーポリゴンを配置し、ジャギーブラーを生成する。ジャギーブラーの色は3.1.4節で求めた色とする。ジャギーブラーは相対速度 $\mathbf{B}$ の逆方向に伸び、伸びる長さは相対速度 $\mathbf{B}$ の大きさに応じて変わる。また、後方の輪郭はジャギーブラーを伸ばす長さを長くし、前方の輪郭は短くする。

各3Dモデルの同じ輪郭部分では、ジャギーブラーの伸びる長さが変わらない。

そのため、正弦関数により伸びる長さにはばらつきを持たせる。ジャギーブラーの生成したイメージ図を図 3.7 に示す。

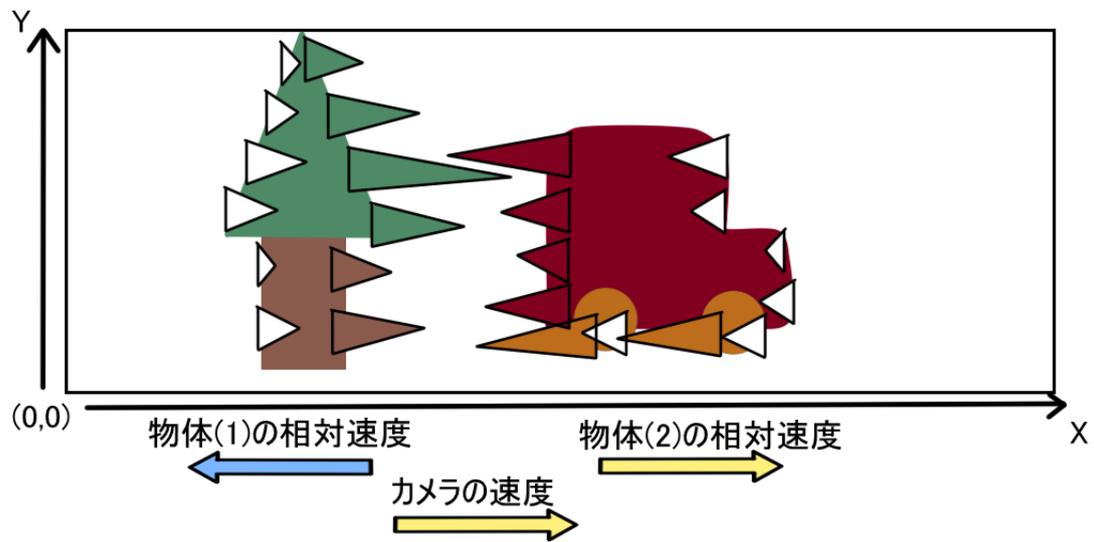


図 3.7: ジャギーブラーの生成

# 第 4 章

## 検証と考察

4.1 節では、カメラとモデルの相対速度によって、見かけ上の動きの前方と後方にそれぞれの特徴を持ったジャギーブラーがでているかを検証する。本手法を検証するために用いるプログラムの実装には、3DCG ツールキットである「Fine Kernel ToolKit」[20]を使用した。4.2 節では、本手法の問題点の考察について述べる。

## 4.1 検証

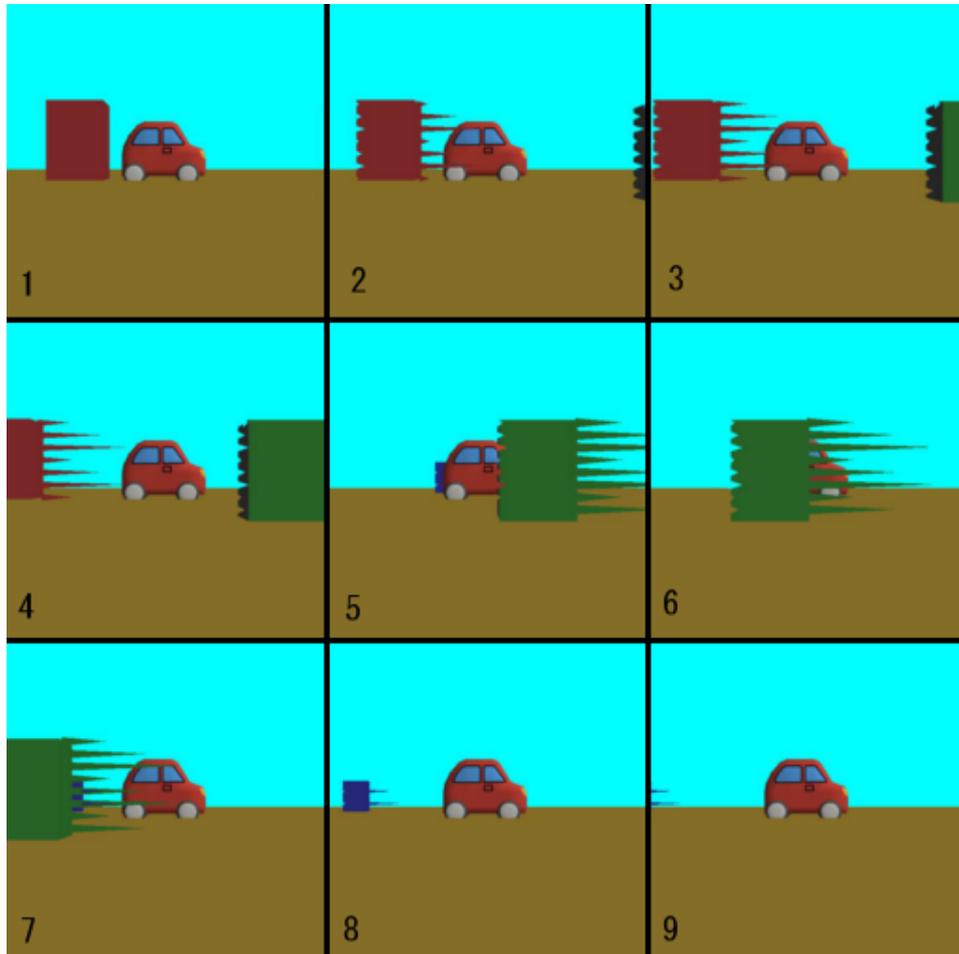
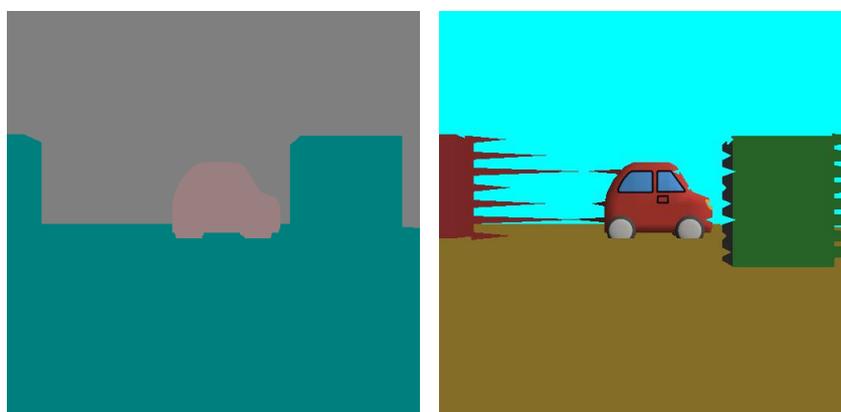


図 4.1: 本手法を用いたジャギーブラーの連番画像

図 4.1 が、本手法を用いて制作した、横方向のジャギーブラーの動画の連番画像である。この動画は、車が停止した状態から右方向に向かって徐々に加速し、その動きに追従しながらカメラも右方向に動く動画である。カメラは車の動きに追従しているため、車の相対速度は 0 である。そのため、車にはジャギーブラーがない。

背景の立方体は動いていないので、背景の相対速度はカメラの移動速度をマイナスした値になる。背景の見かけ上の動きは、左方向に動いて見えるので、立方体の左側の輪郭がへこむように歪み、右側の輪郭が伸びるよう歪む。また、カメ

ラの移動速度が速くなるにつれて立方体の歪みが大きくなる。



(a) 相対速度を色情報で表した画像 (b) ジャギーブラーを描画した画像

図 4.2: 相対速度に応じたジャギーブラーの描画

図 4.2(a)、図 4.2(b) は相対速度に応じたジャギーブラーを描画した図である。この図では、カメラは右方向に動き、車はカメラより速く右方向に動いている。見かけ上の動きが右方向に動くほど R 値が大きい。背景の部分は、R 値が小さくなるので赤の成分が抜け、くすんだシアンの色になる。対して、車の部分は右方向に動いているので、赤色の成分が強い色になる。色情報から見かけ上の動きを判断し、対応した方向にジャギーブラーがでる。

## 4.2 考察

本手法を用いて横方向の動きに対して相対速度から、3DCG で作った背景に手描きアニメのような歪みを実装することができた。

現状の問題点として、輪郭線を持つ 3D モデルに関しては、3D モデルの色ではなく、輪郭線の色で塗るからである。また、見かけ上の動きと実際の動きが異なる場合、実際の動きとは、反対の方向にジャギーブラーがでる。その図を図 4.3 に示す。

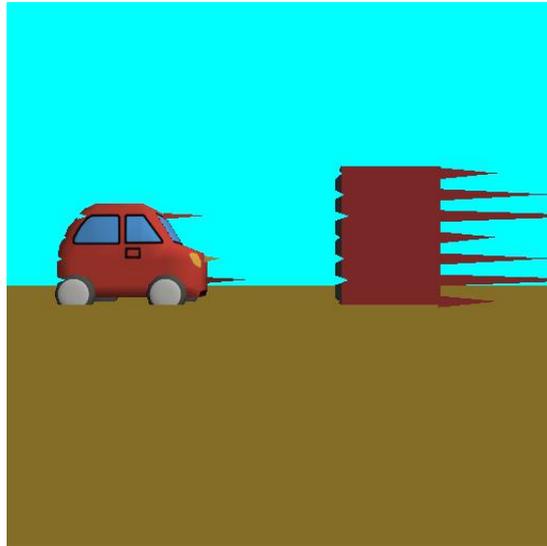


図 4.3: 実際の動きと異なる方向にブラーがでている例

これは相対速度から前方後方の判断するため、カメラの移動速度より遅く動くものはカメラと反対の方向に動いて見えるからである。

これらの問題点については、本研究では解決できなかった。本研究での、車のように主体的な動きをする物体や輪郭線を持つ物体に関しては、本手法とは、別のアルゴリズムを適用する必要がある。また、リアルタイム性が要求されないなら、これらの物体をわけてレンダリングするという方法も有効である。

# 第 5 章

## まとめ

本研究では、3DCG 背景における手描きアニメ調ブラー表現を行うための手法を提案した。カメラとモデルの相対速度からオブジェクトの見かけ上の移動方向における前方の輪郭と後方の輪郭を検出し、前方はへこむように、後方は伸びるようにといった、それぞれの特徴に応じた手描きアニメ調ブラーを表現する事が可能となった。しかし、前方と後方の検出ができない場合や、モデルの実際の動きとは異なる方向にジャギーブラーがでるといった問題があるのでその点を改善していかなければならない。

また、本研究では横方向の動きに焦点を当てたが、手描きアニメでは奥行き方向に動かしたときにもジャギーブラーをだすことがある。奥行き方向の動きにも対応することができれば、より効率的に手描きアニメ調ブラーを表現する事ができるようになるだろう。

# 謝辞

本研究を締めくくるにあたり、まだまだ未熟者の私にときに厳しく、ときに温かくご指導いただきました本校メディア学部の三上浩司先生と渡辺大地先生に心より深く感謝いたします。提案手法を実装するにあたり、多くのアドバイスをくださり、心折れかけたときも適切な助言で支えていただいた竹内亮太先生にも深く感謝いたします。また、研究を進める中で、多くの相談にのっていただいた講師の方々、院生の方々にもお礼を申し上げます。そして、最後に意見を交し合い、苦楽を共にした研究室のメンバーにも心から感謝いたします。皆様本当にありがとうございました。

## 参考文献

- [1] デジタルアニメ制作技術研究会, 東京工科大学. 「プロフェッショナルのためのデジタルアニメマニュアル 2008 ～工程・知識・用語～」. 東京工科大学, 片柳研究所, クリエイティブ・ラボ, pp. 47–70, 1194.
- [2] Amy Gooch, Bruce Gooch, Peter Shirley, Elaine Cohen. "A Non-Photorealistic Lighting Model For Automatic Technical Illustration". *SIG-GRAPH 98*, 1998.
- [3] A.W. Klein, W. Li, M.M. Kazhdan, W.T. Corrêa, A. Finkelstein, and T.A. Funkhouser. "Non-photorealistic virtual environments". In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 527–534. ACM Press/Addison-Wesley Publishing Co., 2000.
- [4] H. Lee, S. Kwon, and S Lee. "Real-time pencil rendering". In *Proceedings of the 4th international symposium on Non-photorealistic animation and rendering*, pp. 37–45. ACM, 2006.
- [5] 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". 東京工科大学メディア学部卒業論文, 2006.
- [6] D.N. Wood, A. Finkelstein, J.F. Hughes, C.E. Thayer, and D.H Salesin. "Multiperspective panoramas for cel animation". In *Proceedings of the 24th annual*

- conference on Computer graphics and interactive techniques*, pp. 243–250. ACM Press/Addison-Wesley Publishing Co., 1997.
- [7] 桑原明栄, 牧野光則. ”CG アニメーション用誇張表現作成補助システムの提案”. 芸術科学会論文誌, Vol. 2, No. 1, pp. 21–30, 2003.
- [8] ビジュアル情報処理編集委員会. ”「ビジュアル情報処理 -CG・画像処理入門-」”. 財団法人 画像情報教育振興協会 (CG-ARTS 協会), p. 128, 2009.
- [9] Yuya KAWAGISHI, Kazuhide HATUYAMA, Kunio KONDO. ”Cartoon Blur: Non-Photorealistic Motion Blur”. *Proceedings of Computer Graphics International 2003*, pp. 276–281, 2003.6.
- [10] Web アニメスタイル -アニメの作画を語ろう 「animator interview うつのみやさとり (3)」 .  
[http://style.fm/as/01\\_talk/utsunomiya03.shtml](http://style.fm/as/01_talk/utsunomiya03.shtml).
- [11] 尾澤直志. 「アニメ作画のきほん ～キャラクター&メカ～」 . ワークスコーポレーション, pp. 146–149, 2010.
- [12] 西川善司の3Dゲームファンのための「CRYSIS」、「CRY ENGINE2」講座知性シェーダーが作り出す新世代グラフィックスの秘密.  
<http://game.watch.impress.co.jp/docs/20080310/3dcry.htm>.
- [13] R.L. Cook, T. Porter, and L Carpenter. ”Distributed ray tracing”. In *ACM SIGGRAPH Computer Graphics*, Vol. 18, pp. 137–145. ACM, 1984.
- [14] SHRED OF GAME -残像とモーションブラー.  
<http://www.platz.or.jp/~moal/mblur.html>.
- [15] 大林正一, 近藤邦雄, 今間俊博. ”3DCG アニメーションのためのカトゥーンブラー”. *Visual Computing グラフィックスとCAD 合同シンポジウム予稿集*, pp. 179–184, 2005.

- [16] 大林正一, 近藤邦雄. "3DCG アニメーションのためのカトウーンブラー". 第67回全国大会講演論文集, Vol. 67(4), pp. 229–230, 2005.
- [17] 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ウーンブラー". IGDA 日本 SIG-GT 第8回「ノンフォトリアリステックにおける生産性の向上へのアプローチ」, 2006.
- [18] Maverick Project -エッジフィルター.  
<http://maverickproj.web.fc2.com/pg41.html>.
- [19] Tomas Akenine-Moller and Eric Haines. リアルタイムレンダリング. ボーンデジタル, 2006.
- [20] Fine Kernel Project. Fine Kernel ToolKit System.  
<http://fktoolkit.sourceforge.jp/>.