

修士論文

平成 25 年度 (2013)

リアルタイム 3DCG における
衝突を考慮したエネルギー波表現に関する研究

東京工科大学大学院
バイオ・情報メディア研究科 メディアサイエンス専攻

仁藤 将輝

修士論文

平成 25 年度 (2013)

リアルタイム 3DCG における
衝突を考慮したエネルギー波表現に関する研究

指導教員 渡辺 大地

東京工科大学大学院
バイオ・情報メディア研究科 メディアサイエンス専攻

仁藤 将輝

論文の要旨

論文題目	リアルタイム3DCGにおける 衝突を考慮したエネルギー波表現に関する研究
執筆者氏名	仁藤 将輝
指導教員	渡辺 大地
キーワード	3DCG, リアルタイムレンダリング, ゲームエフェクト, アニメーション
[要旨]	<p>漫画やアニメーションなどの創作コンテンツにおいてエネルギーの塊が強く発光し、移動していく表現は一般的になった。本研究ではこの表現を「エネルギー波」と呼称し、エネルギー波は3次元空間上に分布する「エネルギー」の集合体とする。エネルギー波は創作表現であり、主に攻撃方法として表現される。エネルギー波の外観は実世界での薄明光線やレーザーと類似している。しかしエネルギー波は他の物体と衝突し、さまざまな形状に変形する。そして衝突後のエネルギー波は間もなく消えていく特徴を持つ。</p> <p>3DCGのビデオゲームにおいてエネルギー波を表現する手法として、少数の矩形ポリゴンにテクスチャアニメーションを利用する手法がある。しかしながら、この手法はエネルギー波の形状を描いたテクスチャ画像をあらかじめ用意する必要がある。そのため、衝突する物体の形状によってエネルギー波の形状を柔軟に変形することはできない。近年ではパーティクルシステムを利用可能な高機能ゲームエンジンが存在する。パーティクルシステムは大量のパーティクルをビルボードテクスチャで描画することにより、炎、煙、雪、雲といった不定形なものを表現することを得意とする。パーティクルシステムを利用し、衝突を考慮したエネルギー波を表現するためには、衝突後のエネルギー波の特性を考慮した新たな運動方程式を定める必要がある。エネルギー波に関する研究は存在するが、リアルタイムで物体との衝突を考慮した研究はない。一方、煙、雲、爆発などをユーザが任意に形状変形できる研究が存在する。しかし実世界には存在しないエネルギー波に対しては適用できない。</p> <p>そこで本研究はリアルタイム3DCGにおいて、エネルギー波の多彩な衝突形状を容易に生成可能な新たな手法を提案する。はじめに衝突後のエネルギー波の形状を形状構成要素として8パターンに分類した。パーティクルシステムと同様にエネルギー波をパーティクルの集合体とし、パーティクルは形状構成要素の8パターンのうちいずれかに所属する。所属したパターンによりパーティクルの運動方程式とレンダリング手法を決定する。パーティクルの所属パターンの決定にはパラメータや確率を利用した。全パーティクルの所属パターンの割合が、最終的なエネルギー波の形状を決定する。提案した手法を実装し、実際のコンテンツとの比較検証をすることにより本研究の有効性を確認した。これにより、多彩なエネルギー波の衝突形状を簡易なパラメータの設定のみで可能とした。</p>

A b s t r a c t

Title	Research on Energy-wave Expression Considering a Collision in Real-time 3DCG
Author	Masaki Nito
Advisor	Taichi Watanabe
Key Words	3DCG, Real-Time Rendering, Game Effect, Animation

[summary]

Expression that moves mass of energy while light emitting strongly became common in creative content, such as animation and comics. This study refers to this expression as "Energy-wave". Energy-wave is an aggregate of "energy" distributed in three-dimensional space. Energy-wave is a creative expression, and is mainly represented as an attack method. Appearance of Energy-wave is similar to the laser and crepuscular rays in the real world. But Energy-wave collides with another object, and shape of Energy-wave is greatly changed by the collision. Energy-wave after the collision has a feature that will soon fade away.

One of a method to express Energy-wave in the video game of 3DCG is a method that uses a texture animation in a rectangular polygon a few. However, it is necessary to prepare in advance a texture image drawn the shape of Energy-wave this approach. Therefore, it is not possible to flexibly deform the shape of Energy-wave by the shape of the object colliding. High-performance game engine available particle systems are present in recent years. By drawing on the Billboard texture of particle mass, particle system is good that you represent anything irregular flame, smoke, snow, and clouds. To represent Energy-wave considering a collision by using a particle system, it is necessary to determine a new equation of motion that takes into account the characteristics of Energy-wave after the collision. Studies on Energy-wave exist, no studies considering collision with an object in real-time. On the other hand, research that allows the user to shape any modifications smoke, clouds, and explosion exists. However, they are not applicable for Energy-wave which do not exist in the real world.

Therefore we propose a new method that can be easily generated a variety of shapes of the Energy-wave after the collision in real-time 3DCG. We were classified into eight shape pattern as a component shape of Energy-wave after the collision at the beginning. Considered a collection of particles Energy-wave as well as particle system, and belongs to one of the eight patterns of shape component particle. Pattern of the components determines the rendering method and the equation of motion of the particle. We were using the probability parameters and the pattern of particle belongs. Percentage of affiliation pattern of all particles determines the shape of Energy-wave. We have confirmed the validity of the study by implement the proposed method and compare the actual content. As a result, we made it possible with a simple parameter setting the various shape of Energy-wave.

目次

第1章	はじめに	1
1.1	研究の背景と目的	2
1.2	論文構成	5
1.3	数式の定義	5
第2章	エネルギー波の衝突表現	7
2.1	表現の多様性	8
2.2	形状構成要素の抽出	8
2.2.1	大域的形状	8
2.2.2	連続性	9
2.2.3	飛沫形状	10
2.2.4	飛沫先端形状	11
2.3	形状構成要素の関係	11
第3章	提案手法	13
3.1	パーティクルシステムを利用した形状生成	14
3.2	パーティクルごとのパターン所属	15
3.3	運動方程式	16
3.3.1	円状	16
3.3.2	不規則状	18
3.4	レンダリング	20
3.4.1	連続型と分離型	20
3.4.2	粒状と棒状	21
3.4.3	丸型と鋭利型	22
第4章	動作検証	24
4.1	実行結果	25
4.2	実行速度の検証	35
第5章	おわりに	36
	謝辞	38

目 次

1.1	代表的なエネルギー波の例	2
1.2	衝突後のエネルギー波のさまざまな形状	3
2.1	大域的形状の2つの要素	9
2.2	連続性の2つの要素	10
2.3	飛沫形状の2つの要素	10
2.4	飛沫先端形状の2つの要素	11
2.5	形状構成要素の関係図	12
3.1	パーティクル分布によるレンダリング	14
3.2	パーティクルの運動方程式とレンダリング手法の決定	15
3.3	物体と衝突時のパーティクルの加速度 A の向き	16
3.4	反発力 R の模式図	17
3.5	制御点を用いた場合のレンダリング結果例	19
3.6	連続型のレンダリング	21
3.7	棒状のレンダリング	22
3.8	鋭利型のレンダリング	23
4.1	基準となるパラメータ設定でのレンダリング結果	26
4.2	円状分布確率 α の値による形状の比較	27
4.3	連続確率 β の値による形状比較	28
4.4	飛沫サンプリング数 N の値による飛沫形状の比較	29
4.5	飛沫の丸み t の値による飛沫先端形状の比較	30
4.6	他の視点位置からみた様子	31
4.7	2つの物体への衝突	32
4.8	制御点を利用した応用例	33
4.9	実際のコンテンツとの比較結果	34

第 1 章

はじめに

1.1 研究の背景と目的

アニメーションや漫画といった創作コンテンツの中で、エネルギーの塊が強く発光し移動していく表現は一般的になった。実世界では起こりえないこの創作表現をアニメ愛好家などの間では「エネルギー波」と呼称されている。本研究もこれに習い「エネルギー波」と呼ぶこととする。エネルギー波は創作コンテンツ内において主に攻撃方法として表現される。代表的な例として、漫画「ドラゴンボール」[1]における「かめはめ波」や、アニメーション「宇宙戦艦ヤマト」[2]における「波動砲」が挙げられる。図 1.1 に漫画「ドラゴンボール」における「かめはめ波」の例を示す。



出典: “ドラゴンボール”

©鳥山明、集英社

図 1.1: 代表的なエネルギー波の例

本研究でのエネルギー波の定義については阿部ら [3][4] の定義と同様とする。すなわち、「空間中のエネルギーの密度が高い場所が強く発光する」「形状変化を伴いながらある地点に向かって移動する」現象と定義する。定義中の「エネルギー」とは、創作コンテンツ内において空間中に存在するエネルギー波を構成する要素で、3次元空間上に分布し、密度が高い部分が強く発光するものとする。

エネルギー波の外観は、実世界での光の道筋が観測できる現象である薄明光線

やレーザーと類似している。しかしエネルギー波は光とは異なり、光速で移動しない。またエネルギー波は他の物体と衝突し、衝突により形状が大きく変化する。そして衝突後のエネルギー波は間もなく消えていく特徴を持つ。これらの点が実世界における薄明光線やレーザーと異なる。

エネルギー波の衝突後の形状は創作コンテンツの作品ごとにさまざまに表現される。図 1.2 は実際の創作コンテンツ [5][6][7] での衝突後のエネルギー波の形状表現の一例である。



出典:“ドラゴンボールZ
危険なふたり!超戦士はねむれない”
©東映



出典:“宇宙戦艦ヤマト 2199”
©XEBEC、AIC、宇宙戦艦ヤマト
2199 製作委員会



出典:“蒼き鋼のアルペジオ -アルス・ノヴァ-”
©Ark Performance、サンジゲン、アルペジオ・パートナーズ

図 1.2: 衝突後のエネルギー波のさまざまな形状

3DCG のビデオゲームにおいてエネルギー波を表現する手法として、少数の矩形ポリゴンにテクスチャアニメーションを利用する手法がある。この手法は処理負荷が軽く、リアルタイム性に優れているため多くのビデオゲームで用いられている。しかしながら、この手法はエネルギー波の形状を描いたテクスチャ画像をあらかじめ用意する必要がある。そのため、衝突する物体の形状によってエネルギー

波の形状を柔軟に変形することはできない。近年ではエネルギー波などのゲームエフェクトの表現に3Dエフェクト用ミドルウェアであるBISHAMON[8]や、高性能ゲームエンジンであるUnity[9]を利用するゲームが増えている。これらのミドルウェアやゲームエンジンを利用しゲームエフェクトを表現する際は、従来の手法に加え、パーティクルシステム[10][11]を利用することが多い。パーティクルシステムは大量のパーティクルをビルボードテクスチャなどで描画することにより、炎、煙、雪、雲といった不定形なものを表現することを得意とする。パーティクルシステムを利用し、衝突を考慮したエネルギー波を表現するためには、衝突後のエネルギー波の特性を考慮した新たな運動方程式を定める必要がある。

エネルギー波の表現に関する研究として、阿部らはエネルギー波のエネルギー分布を解析的線積分可能な関数で定めエネルギー波を表現した。これにより任意視点において正確なエネルギー波表現を可能としたが、エネルギー波と他の物体との衝突に関しては考慮していない。仁藤ら[12]はエネルギー波の発光により発生する光源効果を、擬似的に点光源を配置することにより実現した。しかしながらエネルギー波が他の物体へ与える影響のうち、光源効果のみを実現しており、衝突に関しては考慮していない。

実世界における光跡の表現に関する研究[13][14]や、グレア効果の表現に関する研究[15][16]が存在する。これらの研究はエネルギー波のように物体との衝突や形状変形が起こる現象に適用することはできない。Nowrouzezahraiら[17]はフォトンマッピングのアルゴリズムを利用し、アーティストが容易にボリュームライトを生成できるシステムを開発したが、このシステムはリアルタイム3DCGには不向きである。

煙、雲、爆発などの自然現象をシミュレーションする研究[18][19][20]は多数存在するが、流体の自然な動きを保ちつつ流体の動きをユーザが制御しさまざまな流体形状を生成する研究がいくつかある。Treuilleら[21]やFattalら[22]はユーザが指定した形状へ煙が形状変化するアニメーション生成手法を提案した。Dobashiら[23]はユーザが指定した形状に一致するように動く積乱雲のシミュレーション

を提案した。佐藤ら [24] は複雑なパラメータを必要とせず、ユーザが指定したステップ数で指定した形状に形状変化する爆発シミュレーションを提案した。これらの研究は実世界における煙、雲、爆発などの現象をもとにシミュレーションを行っているため、実世界には存在しないエネルギー波に対しこれらの手法を直接適用することはできない。

そこで本研究では阿部らの定義に加え、「エネルギー波は他の物体との衝突が発生し、衝突によりエネルギー波の形状は変化する」「衝突後のエネルギー波は間もなく消えていく」という定義を追加する。そして、リアルタイム 3DCG において簡易なパラメータの設定のみでエネルギー波の多彩な衝突表現を可能にする新たな手法を提案する。まず衝突後のエネルギー波の形状を形状構成要素として 8 パターンに分類する。パーティクルシステムと同様にエネルギー波をパーティクルの集合体とし、パーティクルは形状構成要素の 8 パターンのうちいずれかに所属する。所属したパターンによりパーティクルの運動方程式とレンダリング手法を決定する。パターンの所属にはパラメータや確率を利用する。パーティクルの所属する割合により最終的なエネルギー波の形状を決める。提案した手法を実装し、実際のコンテンツとの比較検証をすることにより本研究の有効性を確認した。これにより、多彩なエネルギー波の衝突形状を簡易なパラメータの設定のみで可能とした。

1.2 論文構成

本論文は全 5 章で構成する。第 2 章でエネルギー波の衝突表現の多様性とその分析について述べる。第 3 章で提案手法について述べ、第 4 章において提案手法の実装結果を検証した。第 5 章でまとめと今後の展望を述べる。

1.3 数式の定義

本論文で用いる数式を次で定義する。

- $A \cdot B$ はベクトル A と B の内積を表す。
- $|V|$ はベクトル V のノルムを表す。

第 2 章

エネルギー波の衝突表現

本章ではエネルギー波の衝突表現を分析する。2.1 節では表現の多様性について述べる。2.2 節ではエネルギー波の衝突時の形状を分析し、形状を構成する要素を抽出した。2.3 節では 2.2 節で抽出した要素の関係について述べる。

2.1 表現の多様性

エネルギー波の衝突表現は創作コンテンツの世界観や制作者によりさまざまである。同じ作品内であってもシーンごとにエネルギー波の衝突表現が異なることも多い。しかし、エネルギー波の性質を考慮することにより、エネルギー波の形状をある程度分類することができる。エネルギー波が実世界の光と似た性質を持つ場合、エネルギー波の動きは直線的となり、グレア効果を表すような放射状へ鋭く広がる形状や、熱により火花が散っているような形状となる。エネルギー波が流体と似た性質を持つ場合、物体との衝突時には多数の飛沫が飛散したような形状や、流体の質量により衝撃波が発生しているような形状となる。エネルギー波は創作の現象であるが、実世界における複数の現象を模倣しているため表現に多様性が生じていると考えることができる。

2.2 形状構成要素の抽出

本研究ではエネルギー波のさまざまな衝突表現について、制作年代や作品の傾向を問わず、漫画、2D アニメーション、3D アニメーション、特殊撮影、ビデオゲームにおいて、約 50 作品、約 150 シーンを基に調査した。調査の結果、本研究ではエネルギー波の形状は 4 つの形状構成要素により成り立つと分析した。これら 4 つの形状構成要素はそれぞれ、2 パターンずつ典型的な形状を持つと考えた。

2.2.1 大域的形状

第 1 の形状構成要素は大域的形状である。大域的形状とは、エネルギー波全体の大域的な形状を示すもので、大域的にみてエネルギー波が規則的に円状に広がっ

ているか(円状)、規則性はなく不規則状に広がっているか(不規則状)の違いを持つ。図 2.1 は大域的形状の 2 つの要素の違いを表す模式図である。

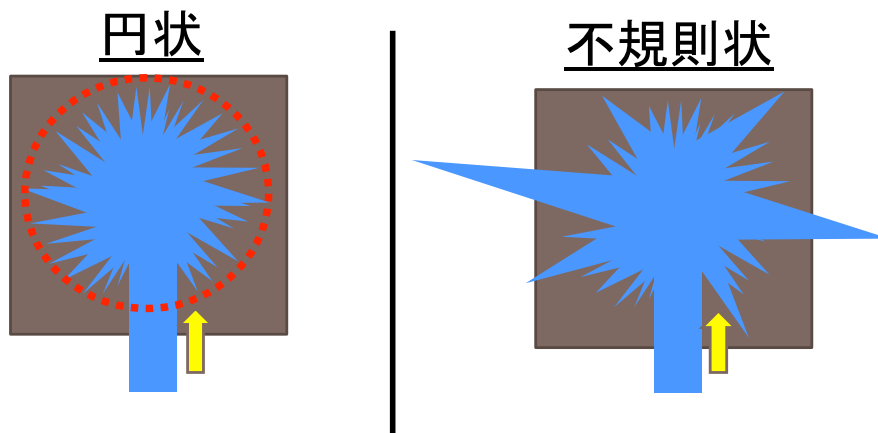


図 2.1: 大域的形状の 2 つの要素

2.2.2 連続性

第 2 の形状構成要素は連続性である。連続性とは、エネルギー波の衝突後、衝突の勢いで飛び散ったエネルギーが衝突した位置から分離せずつながっており、連続しているかを表す。連続しているならば連続型、分離しているならば分離型とする。図 2.2 は連続性の 2 つの要素の違いを示す模式図である。

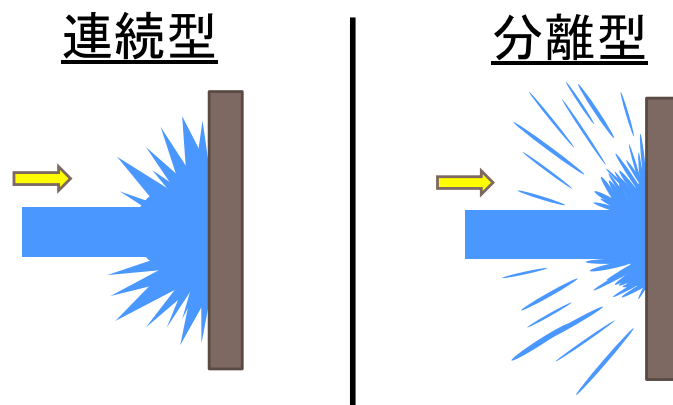


図 2.2: 連続性の 2 つの要素

2.2.3 飛沫形状

第 3 の形状構成要素は飛沫形状である。飛沫形状とは、連続性が「分離型」であった場合のエネルギーひと塊の形状である。飛沫形状は、粒状か棒状かの違いを持つ。図??は飛沫形状の 2 つの要素の違いを表す模式図である。

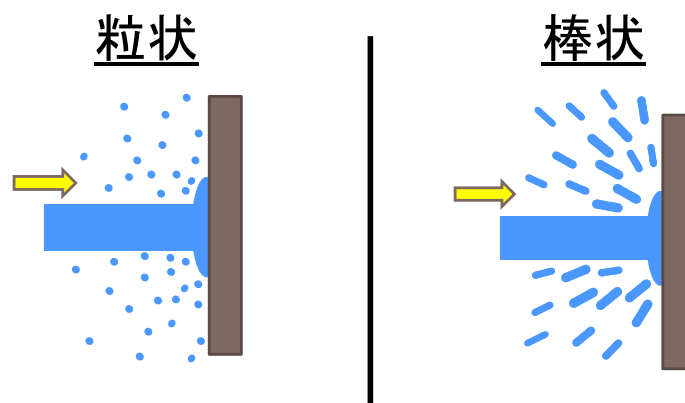


図 2.3: 飛沫形状の 2 つの要素

2.2.4 飛沫先端形状

第4の形状構成要素は飛沫先端形状である。飛沫形状が「棒状」であった場合のエネルギーひと塊の先端が丸型か鋭利型かの違いを持つ。図2.4は飛沫先端形状の2つの要素の違いを表す模式図である。

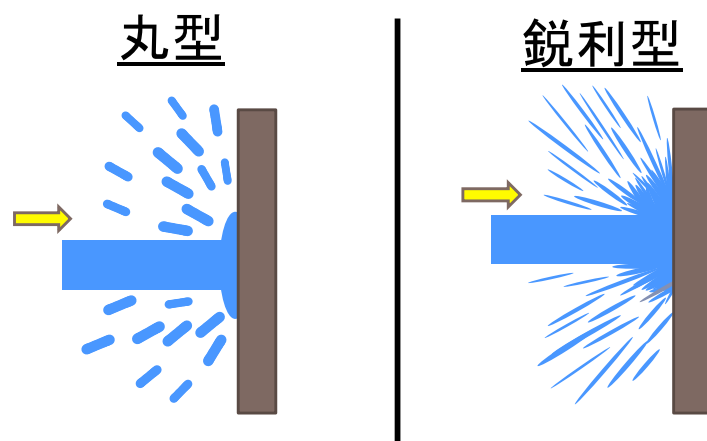


図 2.4: 飛沫先端形状の2つの要素

2.3 形状構成要素の関係

前節で述べた形状構成要素の関係をまとめると8パターンに分類できる。分類した8パターンをそれぞれA、B、C、D、E、F、G、Hとし、図2.5に形状構成要素の関係を示した。

		円状	不規則
	連続型	A	B
分離型	粒状	C	D
	棒状	丸型	F
		鋭利型	G
		E	F

図 2.5: 形状構成要素の関係図

ただし、実際のコンテンツ中でのエネルギー波は、これら 8 パターンのうち、いくつかのパターンが複合する場合もある。また飛沫形状や飛沫先端形状は定性的な要素であるため、どちらに近いかで分類することとなる。

第 3 章

提案手法

本章ではエネルギー波の多彩な衝突形状の表現手法を提案する。3.1節にてパーティクルシステムの利用について述べる。3.2節ではエネルギー波をレンダリングするためのパーティクルごとに、パターンの所属を行う。3.3節ではパーティクルが所属したパターンごとの運動方程式について述べる。3.4節ではパーティクルが所属したパターンごとのレンダリング手法について述べる。

また本章以降のパラメータとはユーザ入力値を指すこととする。パラメータはユーザが変更しない限り値が変わることはない。

3.1 パーティクルシステムを利用した形状生成

パーティクルシステムと同様にエネルギー波をパーティクルの集合体として扱う。基本的な方針としてパーティクルをエネルギー波の形状に分布させ、パーティクルの位置にテクスチャを加算合成し、常にテクスチャを視点方向へ向けることでエネルギー波の形状をレンダリングする。テクスチャは円状にエネルギーが分布しているものを利用することで、いずれの視点から見ても3次元空間上のエネルギーの分布にあわせエネルギー波の形状をレンダリングすることができる。図3.1はパーティクル分布によるレンダリングの模式図である。

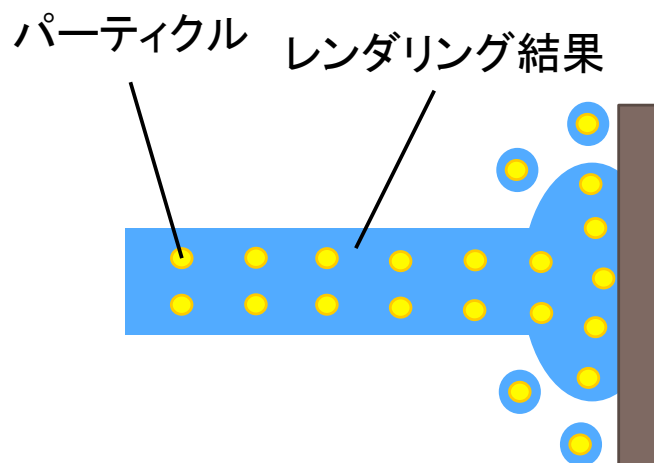


図 3.1: パーティクル分布によるレンダリング

3.2 パーティクルごとのパターン所属

すべてのパーティクルは2章で抽出した形状構成要素の8パターンのうち、いずれか1つのパターンに必ず所属する。パーティクルは所属するパターンごとに運動方程式とレンダリング手法が決まる。大域的形状のどちらかの所属によりパーティクルの運動方程式が決定する。連続性、飛沫形状、飛沫先端形状のいずれかの所属によりパーティクルのレンダリング手法が決定する。図3.2はパーティクルが所属パターンごとに運動方程式とレンダリング手法を決定することを表した図である。

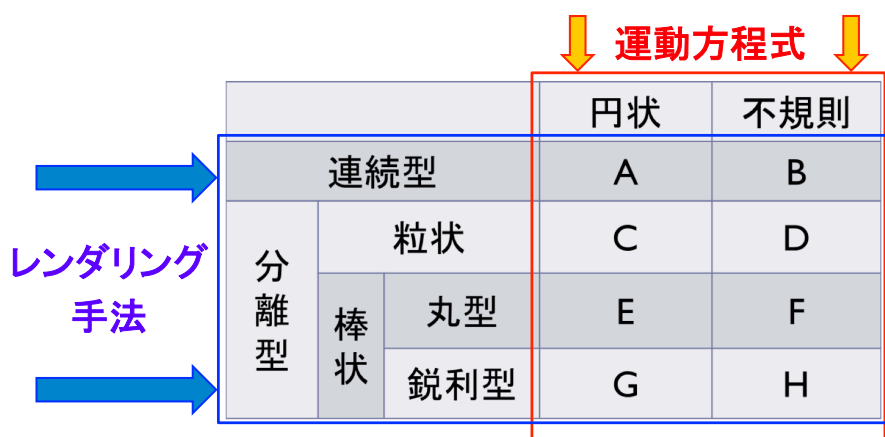


図 3.2: パーティクルの運動方程式とレンダリング手法の決定

エネルギー波を構成するパーティクルがどのパターンに所属するかの割合により、最終的なエネルギー波の形状が決まる。

パーティクルの所属パターンの決定にはユーザが定めるパラメータと確率を利用する。円状か不規則状かの決定には円状分布確率を用いる。円状分布確率の詳細は次節で述べる。連続型か分離型かの決定には連続確率を用いる。連続確率の詳細は3.4.1節で述べる。粒状か棒状かの決定には飛沫サンプリング数を用いる。飛沫サンプリング数の詳細は3.4.2節で述べる。丸型か鋭利型かの決定には飛沫の丸みを用いる。飛沫の丸みの詳細は3.4.3節で述べる。

3.3 運動方程式

パーティクルの運動方程式は所属パターンが円状であるか、または不規則状であるかで大きく異なる。所属パターンの決定にはパラメータ円状分布確率 $\alpha (0 \leq \alpha \leq 1)$ を用いる。 α の値により式 (3.1) のように確率的に決まる。

$$\begin{cases} \alpha & \cdots \text{円状に所属する確率} \\ 1 - \alpha & \cdots \text{不規則状に所属する確率} \end{cases} \quad (3.1)$$

式 (3.1) で決定した所属パターンに従い、パーティクルの運動方程式を求める。

3.3.1 円状

所属パターンが円状のパーティクルは、大域的にみて円状にパーティクルが分布するように運動しなければならない。そのため本手法では物体からの抗力 D と周囲のパーティクルからの反発力 R を導入する。この2つの力により物体と衝突時のパーティクルの加速度 A の向きを決める。図 3.3 が加速度 A の向きを求める模式図である。

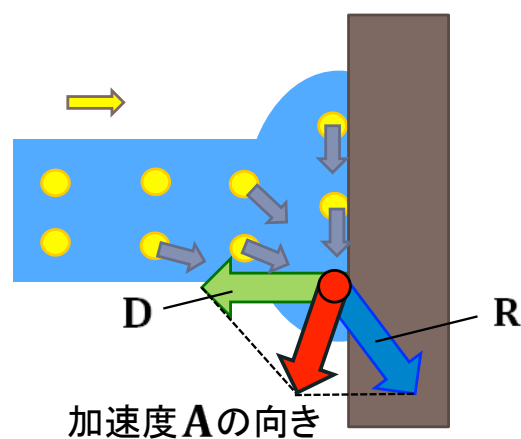


図 3.3: 物体と衝突時のパーティクルの加速度 A の向き

以下に D の求め方を述べる。物体からの抗力にはペナルティ法を用いる。ペナルティ法は物体間の多少の侵入を許し、進入した量に応じた力を抗力とする。バ

ネ係数を k 、ダンパ係数を h 、パーティクルが衝突する物体に侵入した距離を z 、衝突する物体の面の法線方向を \mathbf{n} 、パーティクルの速度を \mathbf{w} とすると D は式 (3.2) で求める。

$$\mathbf{D} = kz\mathbf{n} + h(\mathbf{w} \cdot \mathbf{n})\mathbf{n} \quad (3.2)$$

次に反発力 \mathbf{R} の求め方を述べる。先述通り、反発力は周囲のパーティクルから受けると考える。本手法は粒子法の1つである SPH 法 [25][26] を参考とした。パーティクルは一定の有効半径を持ち、有効半径内に存在する他のパーティクルから力を受けることとする。ただし他のパーティクルが同位置に存在した場合は力を考えない。反発力を受けるパーティクルの有効半径内に存在する他のパーティクル数を n 、反発力を受けるパーティクルの位置を \mathbf{P} 、有効半径内の他のパーティクルの位置を \mathbf{Q}_i 、距離 1 のときの反発力を r としたとき、周囲のパーティクルから受ける力の合計 \mathbf{R} は式 (3.3) で求める。

$$\mathbf{R} = r \sum_{i=1}^n \frac{\mathbf{Q}_i - \mathbf{P}}{|\mathbf{Q}_i - \mathbf{P}|^2} \quad (3.3)$$

図 3.4 が \mathbf{R} を求める模式図である。

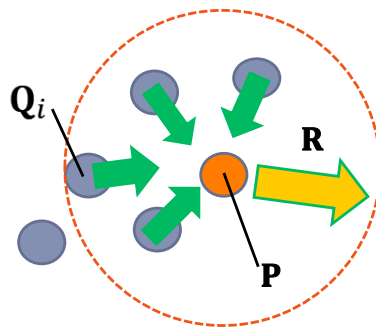


図 3.4: 反発力 \mathbf{R} の模式図

最後に D と \mathbf{R} 、係数 c により \mathbf{A} を求める。 \mathbf{A} は式 (3.4) により求める。

$$\mathbf{A} = \frac{\mathbf{D} + \mathbf{R}}{|\mathbf{D} + \mathbf{R}|} c \quad (3.4)$$

式(3.4)によりパーティクルごとに \mathbf{A} の向きは異なるが、大きさは同じとなる。すべてのパーティクルの \mathbf{A} の大きさは c により決定する。これにより、パーティクルは大域的にみて円状に分布する。なおパーティクルの寿命の値により、衝突後パーティクルが消滅するまでの時間は決まる。本節では円状の運動方程式について述べているが、 c やパーティクルの寿命に対しランダム幅を持たせ、値にばらつきを与えることで不規則状のパーティクル分布を生成することも可能である。

3.3.2 不規則状

不規則状に分布しようとするパーティクルに対しては、制御点を用いる。本研究は見た目の制御を重視するため、よりユーザが直接的に制御しやすい制御点を用いることとした。パーティクルは物体と衝突後、制御点へ向かって移動し、制御点の位置で寿命が尽き消滅する。制御点は1つのエネルギー波に対しいくつか事前に設定する。制御点は3次元空間上の絶対的な位置ではなく、エネルギー波ごとの相対的な方向と相対的な位置を表す。制御点が複数存在する場合は制御点ごとに確率を設定し、各パーティクルがどの制御点に移動していくかを確率で求める。図3.5は制御点を3つ設定し、パーティクルが3方向へ枝分かれするレンダリング結果の例である。

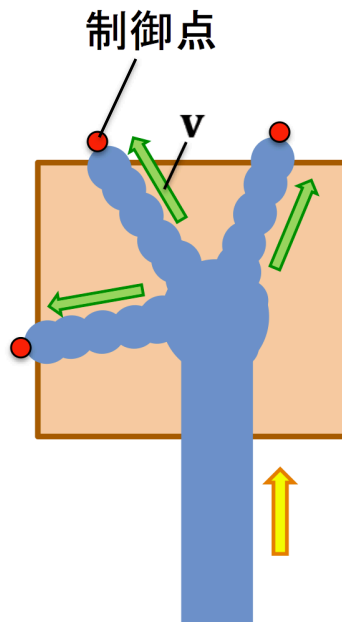


図 3.5: 制御点を用いた場合のレンダリング結果例

パーティクルの衝突位置を H 、制御点の位置を E 、パーティクルが衝突してから消滅するまでの時間である寿命を τ としたとき、パーティクルの衝突後の速度 v は式 (3.5) で求める。

$$v = \frac{E - H}{\tau} \quad (3.5)$$

制御点の位置を変えない場合、エネルギー波は終始ほぼ同じ形状のままである。しかし任意の時間間隔でいくつかの制御点を移動させたり、新たに制御点を配置したりすることで、キーフレームアニメーションのようにエネルギー波の形状を時間毎に変形することも可能である。また Perlin noise[27] を利用し毎フレーム制御点をランダムに移動することでより自然な動きのアニメーションとなりやすい。制御点の利用によりユーザは容易に不規則状のパーティクルの分布が可能になるが、制御点の配置位置によっては円状のパーティクル分布を生成することも可能である。

3.4 レンダリング

前節ではパーティクルの運動方程式について述べたが、本節ではパーティクルの所属パターンごとのレンダリング手法について述べる。レンダリング手法の所属パターンは、はじめに連続型か分離型のどちらかに決まり、次いで粒状か棒状かが決まり、最後に丸型か鋭利型かが順番に決まる。

なお衝突後のパーティクルはすべて残りの寿命に応じて透明度を下げて描画することでエネルギー波が自然に消滅していくように見える。

3.4.1 連続型と分離型

連続型と分離型かの決定にはパラメータ連続確率 $\beta (0 \leq \beta \leq 1)$ を用いる。3.3節の円状分布確率と同様に、次の式 (3.6) により確率的に決まる。

$$\begin{cases} \beta & \cdots \text{連続型に所属する確率} \\ 1 - \beta & \cdots \text{分離型に所属する確率} \end{cases} \quad (3.6)$$

分離型の場合は次節の粒状か棒状のレンダリング手法に従う。

連続型の場合は本節のレンダリング手法に従う。連続型のレンダリングでは、各パーティクルは衝突した位置を各々記憶しておく。衝突した位置から現在のパーティクル位置に向けテクスチャを追加して配置し描画することで連続型を表現する。1つのパーティクルに対し M 個のテクスチャを利用するとし、パーティクルの衝突位置を P_0 、パーティクルの現在位置を P_1 、をとすると $i (i = 1, 2, 3 \dots M)$ 番目に表示するテクスチャ位置 T_i は式 (3.7) で求める。

$$T_i = P_0 + \frac{P_1 - P_0}{M} (i - 1) \quad (3.7)$$

また連続型ではパーティクルが現在位置に近づくほどテクスチャサイズの倍率を下げて描画する。これにより連続型の形状を表現する。図 3.6 がテクスチャ配置を利用した連続型のレンダリングの模式図である。

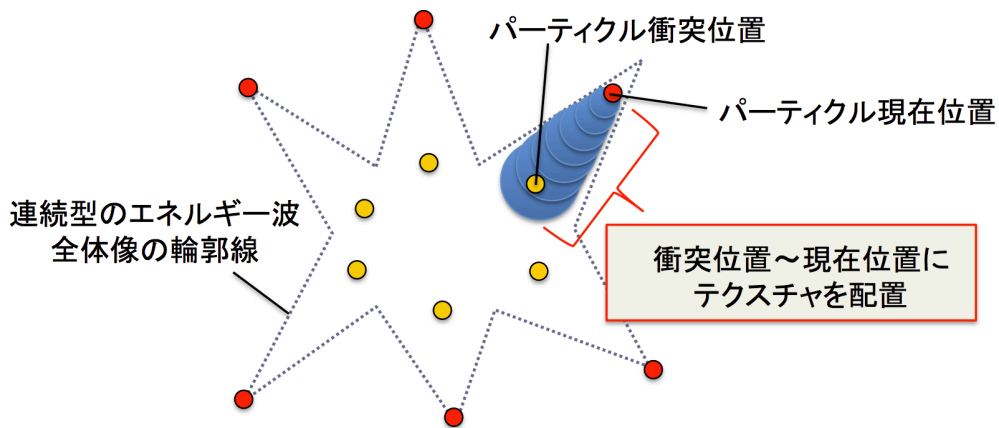


図 3.6: 連続型のレンダリング

3.4.2 粒状と棒状

本節では、分離型の場合に生じる各飛沫形状のレンダリングについて述べる。飛沫のサンプリング数という整数のパラメータ N を利用する。 $N = 1$ ならば粒状、 $1 < N$ ならば棒状となる。

粒状の場合は単純にパーティクルの位置に 1 つのテクスチャを表示する。

棒状の場合は粒状の表示も含み、パーティクル進行方向両側に N 個のテクスチャを配置し描画する。パーティクルの位置を \mathbf{Q} 、パーティクルの進行方向単位ベクトルを \mathbf{d} 、テクスチャ表示間隔を u で表したとき、 j ($j = 1, 2, 3, \dots, N$) 番目に表示するテクスチャ位置 \mathbf{U}_j は式 (3.8) で求める。

$$\mathbf{U}_j = \mathbf{Q} + (-1)^j u(j - 1)\mathbf{d} \quad (3.8)$$

これにより棒状の表現が可能である図 3.7 がテクスチャの追加配置を利用した棒状の表現の模式図である。

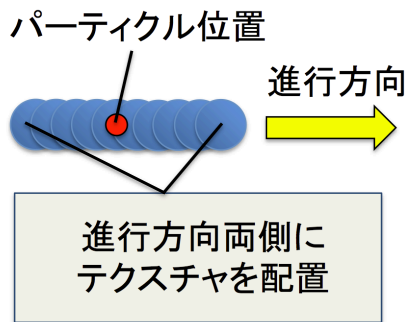


図 3.7: 棒状のレンダリング

レンダリング手法が棒状の場合はさらに次節のレンダリング手法にも従う。

3.4.3 丸型と鋭利型

レンダリング手法が棒状の場合、飛沫の丸みというパラメータ $t(0 < t \leq 1)$ により飛沫先端形状を表現する。 t の値を描画するテクスチャサイズに乘算し、乗算時は先端のテクスチャほど t の影響力を強める。これにより、 t が 1 に近い値であるほど丸型となり、 t が 0 に近い値であるほど鋭利型となる。 j 番目に表示するテクスチャサイズ S_j は式 (3.9) で求める。ただし、 S はパーティクル位置におけるテクスチャサイズとする。

$$S_j = \frac{N - (j - 1)(1 - t)}{N} S \quad (3.9)$$

これにより鋭利型を表現可能である。図 3.8 が鋭利型にテクスチャサイズを変更する模式図である。

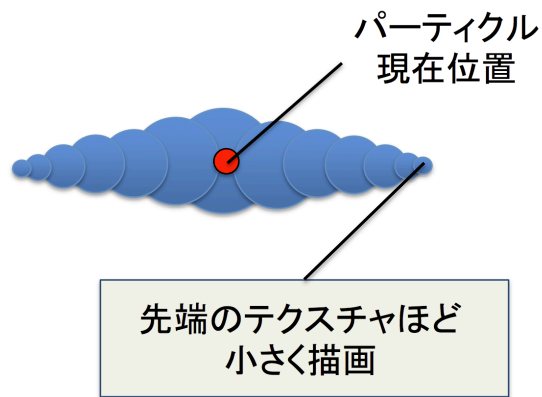


図 3.8: 鋭利型のレンダリング

第 4 章

動作検証

本章では3章で述べた手法を実装する。実装したレンダリング結果から、本手法の有用性を検証する。3DグラフィクスAPIはOpenGL[28]をもとにした3DCGツールキットであるFine Kernel Toolkit System[29][30]を利用した。検証に使用した実行環境を表4.1に示す。

表 4.1: 実行環境

OS	Windows 7 Enterprise 64 ビット
CPU	Intel(R) Core(TM)2 Duo CPU E8500 3.16GHz
GPU	NVIDIA GeForce 9800 GTX/9800 GTX+
メモリ	4GB

4.1 実行結果

提案手法で述べたパラメータや係数の基準値を表4.2のように定め、比較検証を行う。表4.2の上段の4つのパラメータが形状構成要素の8パターンを生成する本手法における重要なパラメータである。図4.1が基準となるパラメータ設定によるレンダリング結果である。

表 4.2: パラメータの基本値設定

円状分布確率	α	1.0
連続確率	β	0.3
飛沫サンプリング数	N	12
飛沫の丸み	t	0.5
パーティクルの寿命	τ	40
衝突時の加速度の大きさ (円状分布-連続型)	c	1.5 ランダム幅 0.5
衝突時の加速度の大きさ (円状分布-分離型)	c	2.5 ランダム幅 1.0
反発力係数	r	4.0
バネ係数	k	1.0
ダンパ係数	h	0.8
棒状のテクスチャ表示間隔	u	0.4

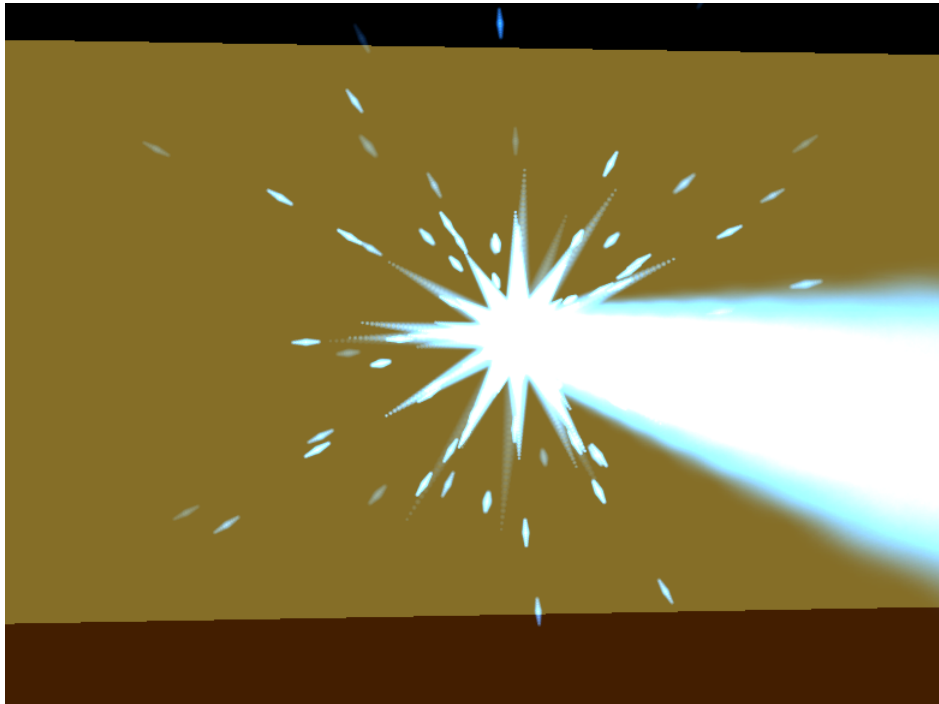
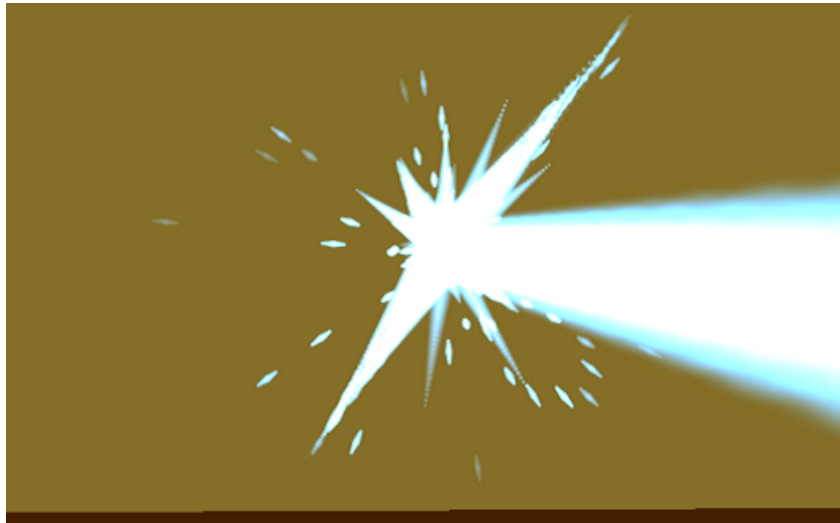
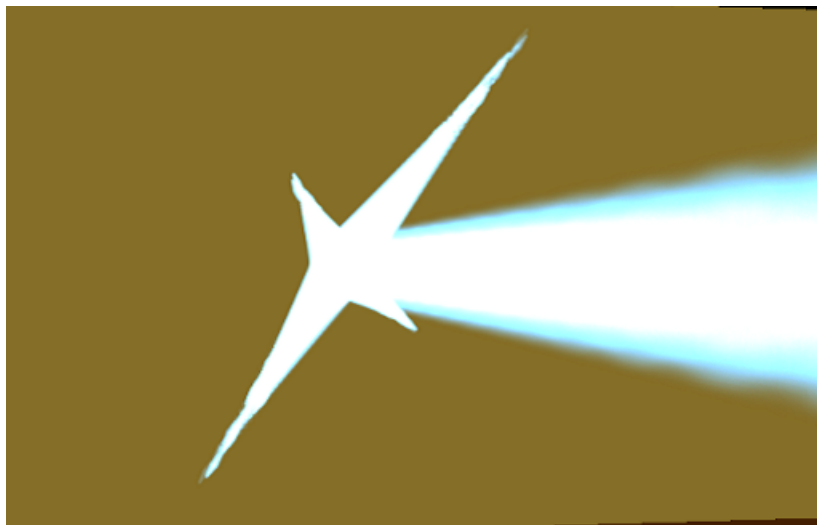


図 4.1: 基準となるパラメータ設定でのレンダリング結果

図 4.2 では基準となるパラメータ設定から円状分布確率 α の値を変更し、大域的形状における円状と不規則状の形状比較を行った。不規則状では制御点を 4 点利用した。



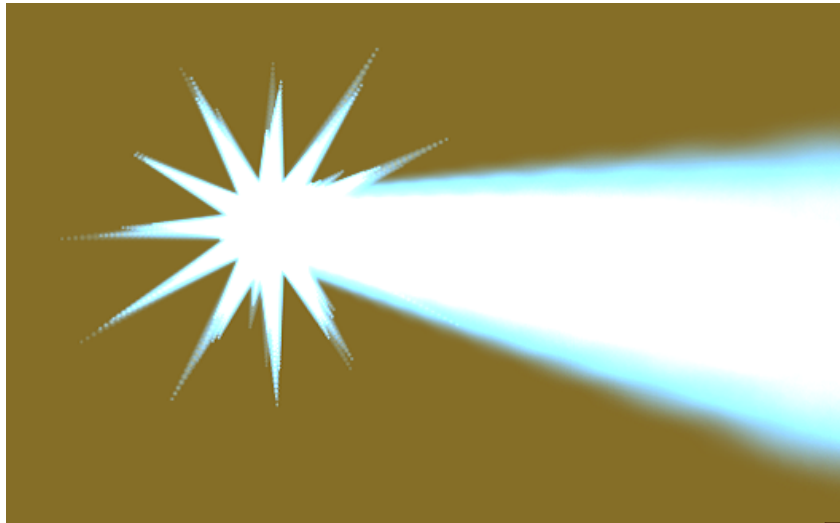
$\alpha = 0.5$ (円状+不規則状)



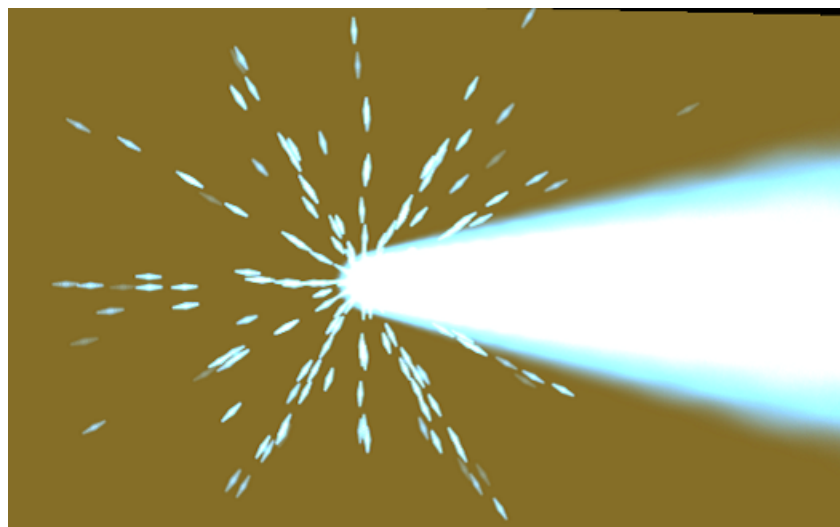
$\alpha = 0.0$ (不規則状)

図 4.2: 円状分布確率 α の値による形状の比較

図 4.3 では基準となるパラメータ設定から連続確率 β の値を変更し、連続性における連続型と分離型のレンダリング結果の比較を行った。



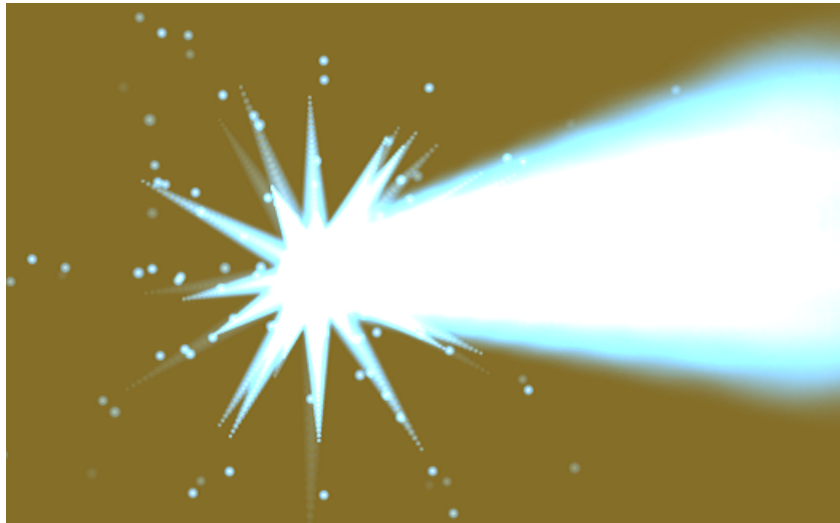
$\beta = 1.0$ (連続型)



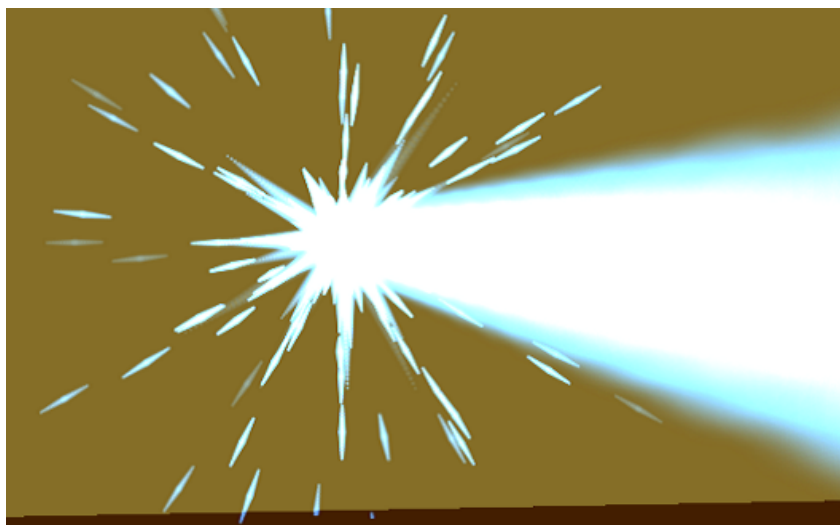
$\beta = 0.0$ (分離型)

図 4.3: 連続確率 β の値による形状比較

図 4.4 は基準となるパラメータ設定から飛沫サンプリング数 N の値を変更し、飛沫形状における粒状と棒状のレンダリング結果の比較を行った図である。



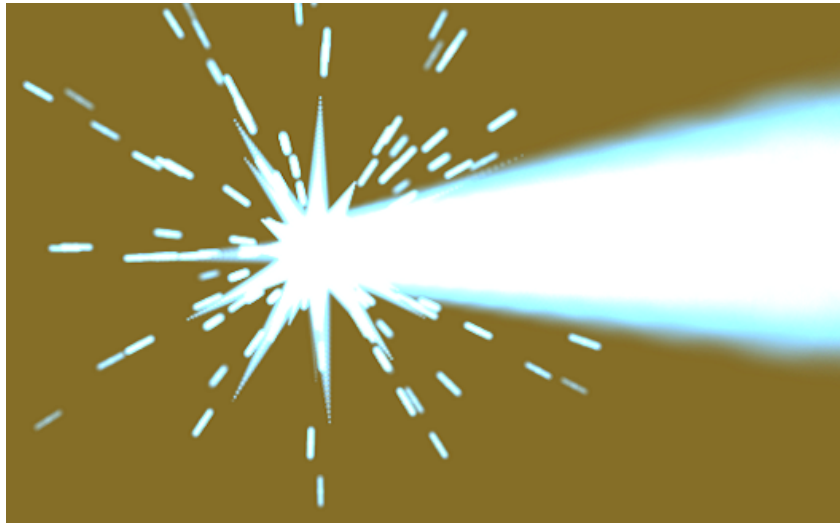
$N = 1$ (粒状)



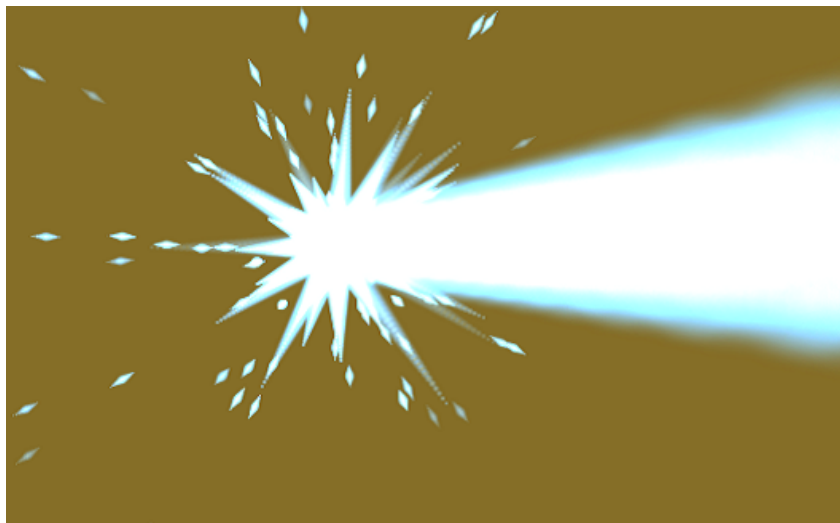
$N = 24$ (棒状)

図 4.4: 飛沫サンプリング数 N の値による飛沫形状の比較

図 4.5 は基準となるパラメータ設定から飛沫の丸み t の値を変更し、飛沫先端形状における丸型と鋭利型のレンダリング結果の比較を行った図である。



$t = 1.0$ (丸型)



$t = 0.1$ (鋭利型)

図 4.5: 飛沫の丸み t の値による飛沫先端形状の比較

図 4.2、図 4.3、図 4.4、図 4.5 から、本研究が提案したパラメータの変更のみで 2 章で分類した 8 つの形状を簡易に生成できることが確認できた。

図 4.6 は基準のパラメータ設定での他の視点位置からの様子である。任意視点に

おいても違和感のないレンダリングが可能であることが確認できる。

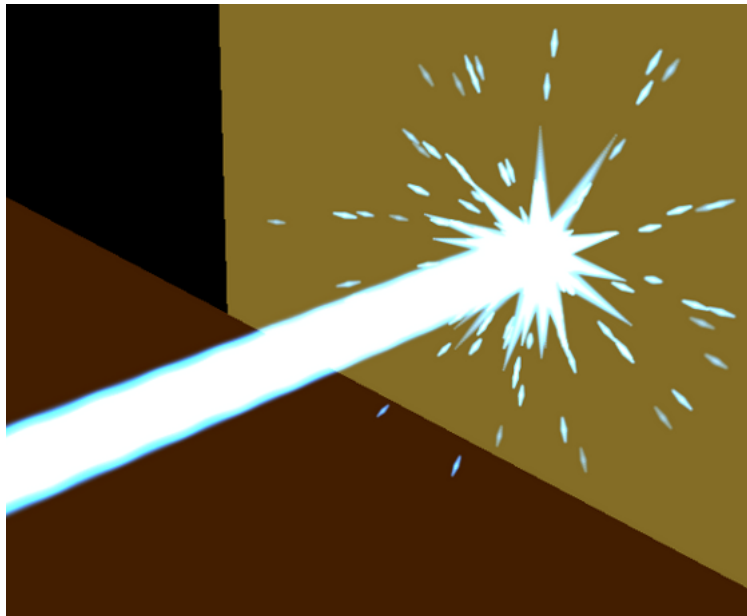
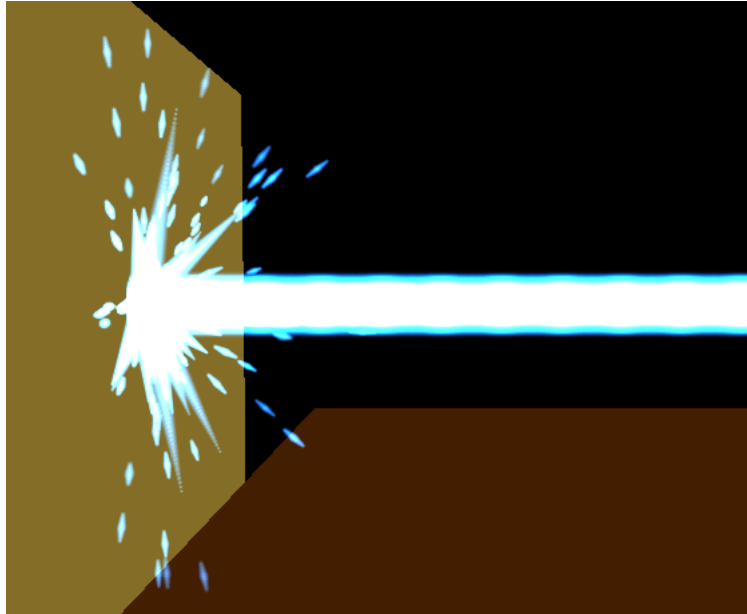


図 4.6: 他の視点位置からみた様子

図 4.7 はエネルギー波の一部が物体に衝突し、衝突しなかった部位が再度異なる

物体に衝突した様子である。

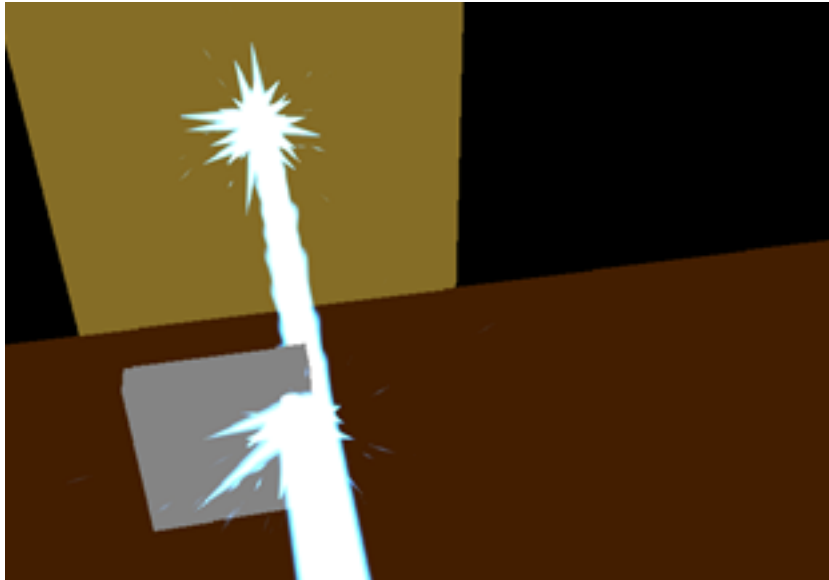


図 4.7: 2つの物体への衝突

図 4.8 ではパーティクルの分布を不規則状とし、制御点の位置へのランダム幅の利用や、制御点の移動によるアニメーション効果を利用した応用例を確認できる。

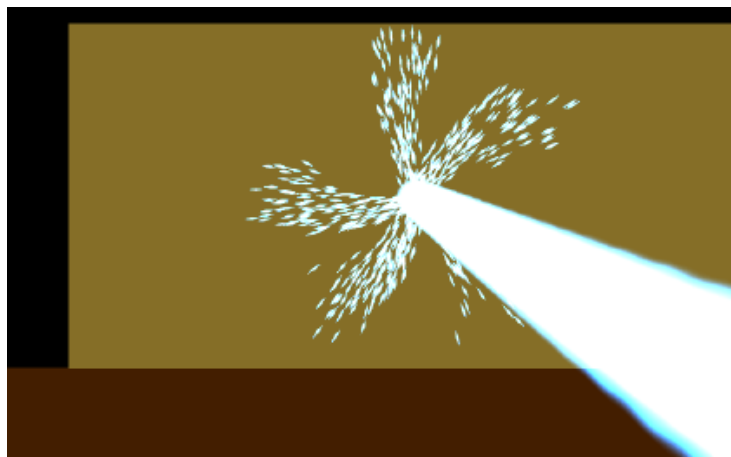


図 4.8: 制御点を利用した応用例

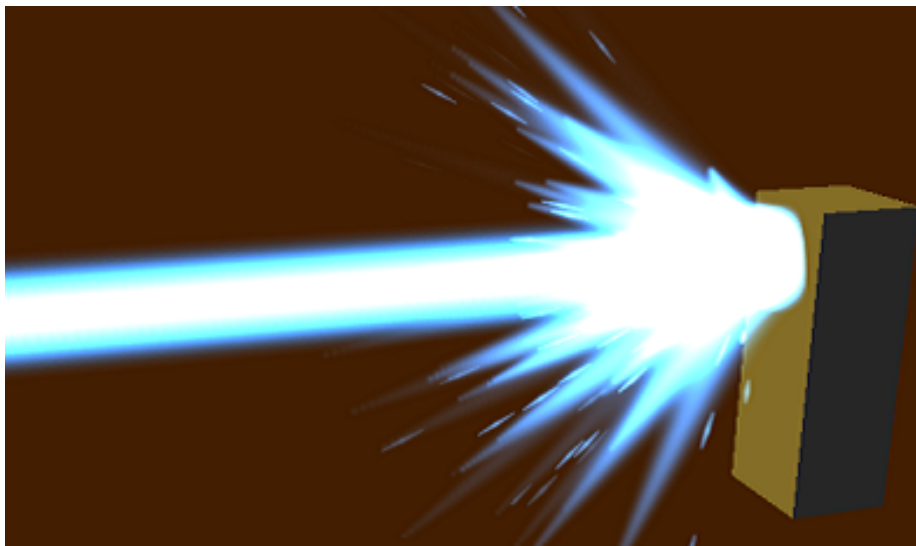
図 4.9 では実際のコンテンツ [31] と本手法との比較結果である。実際のコンテンツに似せるのには 30 分程の時間を要したが、円状分布確率、連続確率、飛沫サンプリング数、飛沫の丸みの 4 つのパラメータを設定し概観を似せるだけならば数分で調整可能であった。主にテクスチャ画像の作成と、バネ係数、ダンパ係数の値の決定に時間を要した。



実際のコンテンツ

出典: “機動戦士ガンダム エクストリームバーサス”

©バンダイナムコゲームス、創通、サンライズ



本手法

図 4.9: 実際のコンテンツとの比較結果

4.2 実行速度の検証

表 4.1 の実行環境でパーティクルの放出を続け、10 秒間の平均フレームレートを計測した。パーティクル数とテクスチャ数を 4 つのケースで計測した結果を表 4.3 に示す。連続型や棒状のレンダリングでは 1 つのパーティクルに対し複数のテクスチャが必要であるため、ケース 3、ケース 4 ではパーティクル 1 に対しテクスチャ数を 5 として計測した。

表 4.3: 処理速度の検証結果

	ケース 1	ケース 2	ケース 3	ケース 4
パーティクル数	1221	2410	1221	2410
テクスチャ数	1221	2410	6105	12050
平均フレームレート	85.1 FPS	31.3 FPS	26.5 FPS	10.4 FPS

第 5 章

おわりに

本研究ではリアルタイム 3DCG において、多彩なエネルギー波の衝突表現を簡易なパラメータ変更のみで生成可能な新たな手法を提案した。

本手法では、創作コンテンツにおける多数のエネルギー波の衝突シーンをもとに形状の分析を行った。分析の結果、形状構成要素を 8 パターン抽出した。パーティクルシステムと同様にエネルギー波をパーティクルの集合体として扱い、パーティクルを形状構成要素の 8 パターンのいずれかに所属させた。パーティクルの運動方程式やレンダリング手法をパターンごとに設定した。パーティクルのパターン所属にはパラメータや確率を利用した。本手法により、多彩なエネルギー波の衝突形状を簡易なパラメータの変更のみで可能とした。調査した 150 シーンのうち 70～80% 程度の形状を本手法で再現が可能であると考えられる。また本論文では直方体への衝突のみを実装したが、パーティクルが物体と衝突しているかさえ判定できればさらに複雑な形状への衝突も適応可能だろう。

今後の課題として、まず現状の手法では表現できないエネルギー波形状への対応がある。エネルギー波が物体と衝突後、流体のように衝突する物体に沿って動いていく形状や、分離したエネルギー波の形状が自由な曲線を描いている場合など、対応が難しい表現もある。また他の課題として、処理の高速化が必要である。パーティクルを大量に利用する場合や、品質の高いレンダリング結果を得るには多くのテクスチャの表示が必要であり、大きな処理負荷がかかる。この課題の解決策の 1 つとして GPGPU やシェーダの利用を考えることができる。ジオメトリシェーダなどを利用することでより高速にテクスチャを表示し、CPU の処理負荷の軽減が期待できる。これらの課題を解決し、多種多様なエネルギー波の衝突形状を高速に表現することができれば、創作コンテンツにおける表現の幅をさらに広げることができるだろう。

本研究は NICOGRAPH2013 において“リアルタイム 3DCG における衝突を考慮したエネルギー波表現”[32] として発表した内容を含む。

謝辭

本論文を締めくくるにあたり、多くの方々への感謝の意を表します。

日常生活から研究の相談、論文執筆、学会発表など、先生方や研究室の間には大変支えられました。ご指導やご協力、ご助言を頂き深く感謝いたします。

霊験灼かな山上に聳え立つ、東京工科大学での日々も終わりです。研究生生活は夢のような日々でしたが、夢をみれない日もありました。本日は数多の思い出と共に布団に入ります。



参考文献

- [1] 鳥山明, 集英社. ドラゴンボール, 1985.
- [2] オフィスアカデミー, 讀賣テレビ. 宇宙戦艦ヤマト, 1974.
- [3] 阿部雅樹, 渡辺大地. エネルギー波表現のリアルタイムレンダリング. 芸術科学会論文誌, Vol. 9, No. 3, pp. 93–101, 2010.
- [4] 阿部雅樹, 渡辺大地. エネルギー波表現のリアルタイムレンダリング. Master's thesis, 東京工科大学大学院バイオ情報・メディア研究科メディアサイエンス専攻, 2010.
- [5] 東映. ドラゴンボールZ 危険なふたり!超戦士はねむれない, 1994.
- [6] XEBEC, AIC, 宇宙戦艦ヤマト 2199 製作委員会. 宇宙戦艦ヤマト 2199, 2013.
- [7] Ark Performance, サンジゲン, アルペジオ・パートナーズ. 蒼き鋼のアルペジオ-アルス・ノヴァ-, 2013.
- [8] マッチロック株式会社. BISHAMON.
<http://www.matchlock.co.jp/>.
- [9] Unity Technologies. Unity.
<http://www.unity3.com/>.
- [10] W. T. Reeves. Particle systems. a technique for modeling a class of fuzzy objects. In *ACM Transactions on Graphics*, Vol. 2, pp. 91–108, 1983.
- [11] E. Haines T. Akenine-Möller. リアルタイムレンダリング 第2版 REAL-TIME RENDERING Second Edition. ボーンデジタル, 2006.
- [12] 仁藤将輝, 渡辺大地. リアルタイム3DCGにおけるエネルギー波の光源効果の表現. In *CEDEC2012*.
<http://cedil.cesa.or.jp/session/detail/980/>, 2012.

- [13] 金沢雄一郎, 牧野光則. レイ・トレーシングによる光跡の表現. 電子情報通信学会総合大会講演論文集, p. 377, 1997.
- [14] 土橋宜典, 山本強, 西田友是. グラフィックスハードウェアを用いた光跡の高精度・高速レンダリング法. 映像情報メディア学会誌, 第52巻, pp. 1–6, 1998.
- [15] K. Zimmerman D. P. Greenberg G. Spencer, P. Shirley. Physically-based glare effects for digital images. In *SIGGRAPH '95 Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pp. 325–334, 1995.
- [16] 柿本正憲, 西田友是, 苗村健, 原島博. グレア効果のヘッドランプ設計検証への応用. 画像電子学会誌, 第39巻, pp. 369–374, 2010.
- [17] A. Selle D. Lacewell D. Nowrouzezahrai, J. M. Johnson and W. Jarosz M. Kaschak. A programmable system for artistic volumetric lighting. In *ACM Transactions on Graphics*. Vol.30, Issue 4, Article No. 29, 2011.
- [18] H. W. Jensen R. Fedkiw, J. Stam. Visual simulation of smoke. In *SIGGRAPH '01 Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 15–22, 2001.
- [19] A. Lastra M. J. Harris. Real-Time Cloud Rendering. In *Computer Graphics Forum*, pp. 76–85. Vol. 20, Issue 3, 2001.
- [20] J. F. O'Brien G. D. Yngve. Animating explosions. In *SIGGRAPH '00 Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 29–36, 2000.
- [21] Z. Popović J. Stam A. Treuille, A. McNamara. Keyframe control of smoke simulations. In *ACM Transactions on Graphics*, Vol. 22, pp. 716–723, 2003.

- [22] L. Dani R. Fattal. Target-driven smoke animation. In *ACM Transactions on Graphics*, Vol. 23, pp. 441–448, 2004.
- [23] T. Nishita T. Yamamoto Y. Dobashi, K. Kusumoto. Feedback control of cumuliform cloud formation based on computational fluid dynamics. In *ACM Transactions on Graphics*. Vol. 27, Issue 3, Article No. 94, 2008.
- [24] 佐藤周平, 土橋宜典, 山本強, 安生健一. 最適な初期強度分布の推定および予測制御による爆発シミュレーションの制御. 映像情報メディア学会誌, 第 65 巻, pp. 1446–1451, 2011.
- [25] L. B. Lusy. A numerical approach to the testing of the fission hypothesis. In *The astronomical journal*, Vol. 82, pp. 1013–1024, 1977.
- [26] M. Gross M. Müller, D. Charypar. Particle-based fluid simulation for interactive applications. In *ACM SIGGRAPH Eurographics symposium on Computer animation*. Eurographics Association, 2003.
- [27] K.Perlin. Improving noise. In *ACM Transactions on Graphics*, Vol. 21, pp. 681–682, 2002.
- [28] OpenGL.org. OpenGL.
<http://www.opengl.org/>.
- [29] Fine Kernel Project. Fine Kernel ToolKit System.
<http://fktoolkit.sourceforge.jp/>.
- [30] 渡辺大地. リアルタイムグラフィックスのためのツールキットに関する研究. PhD thesis, 慶應義塾大学大学院政策 メディア研究科, 1996.
- [31] バンダイナムコゲームス, 創通, サンライズ. 機動戦士ガンダムエクストリームバーサス, 2011.

- [32] 仁藤将輝, 渡辺大地, 柿本正憲, 三上浩司. リアルタイム 3DCG における衝突を考慮したエネルギー波表現. NICOGRAPH2013 論文集, pp. 97–104, 2013.