

エネルギー波表現の
リアルタイムレンダリングに関する研究

2024年9月

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

阿部 雅樹

博士学位論文

エネルギー波表現の
リアルタイムレンダリングに関する研究

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

阿部 雅樹

2024年9月

概要

近年、アニメーションや映画、ビデオゲームといった創作コンテンツ上で3次元コンピュータグラフィックス(以下3DCG)を用いた様々な表現が開発・利用されている。人体の写実的表現や光学効果による場面演出、流体物質の運動等多岐に渡り3DCG技術が活用されている。特に映画やゲームといったコンテンツ上では、エフェクトと呼ばれる視覚効果が用いられている。こうした視覚効果はVisual Effects(VFX)と呼ばれ、コンテンツ内の状況やシーンを彩る手段として発展してきた。映画とゲームはどちらも描画したいシーンを構築した後、実際の映像として描画を行う。映画とゲームではコンテンツの性質上の違いがあり、映画では映像を視聴者へ届ける一方向的なメディアであり、シーンの描画に数分から数時間、あるいは数日かける事が可能である。対してゲームはユーザー側からのインタラクションによって描画すべき映像が変化する双方向的なメディアである。その為、シーンの描画には非常に高速な、1秒間に数十回の映像描画が必要となる。前者の映像描画に多量の時間をかけることができる方針をプリレンダリング、後者の素早く映像描画を行う必要がある方針をリアルタイムレンダリングと呼び、映像制作においては異なる技術体系や考え方となる。

ゲームにおけるエフェクトには、ゲーム内の出来事をユーザーへ伝達する際の伝達効率を向上する説明的役割、シーン内の出来事を装飾し盛り上げる演出的役割が有り、総じてゲームコンテンツの魅力向上をさせる役割である。説明的役割例として、キャラクターが有利な状態へ移行した際に特別な光を発する、ゲーム内UI上で選択しているボタンを強調する等が挙げられる。演出的役割例として、キャラクターが敵に目掛けて魔法を照射する、攻撃が当たったら大きな爆発を付加する、場面全体にカメラの効果を加える等が挙げられる。ゲームエフェクトは大別して2種類ある。ポストプロセス・エフェクトとパーティクル・エフェクトである。ポストプロセス・エフェクトは、画面全体に対して視覚効果を付与する手段である。シーン全体に霧をかけ神妙な雰囲気演出したり、光の当たり方をシミュレートし昼間から夕方にかけての時間変化を一瞬に切り替える演出を行うなどの利用方法がある。パーティクル・エフェクトは、キャラクターや建物といったオブジェクトと同様にゲーム内シーン中に存在する物(オブジェクト)としての視覚効果

である。炎や水・雷といった自然現象、魔法や光線といった架空現象等を、そのシーン中に存在するオブジェクトとして表示する。近年において視覚効果オブジェクトとしてのエフェクトを作成するツールは、パーティクルという粒子を基本単位とする事が多い為、俗にパーティクル・エフェクトと呼ばれている。多量のパーティクルの集合体で流体的なエフェクトそのものを表現する利用方法や、パーティクルの位置座標にアニメーションテクスチャを配置する等、エフェクト素材の発生位置としての利用方法がある。ただし、現在においても矩形ポリゴンや球体といったプリミティブ形状を用いてエフェクトを配置する技法も一般的である。その為、本研究ではこのシーン中に存在する視覚的効果の事を単に「エフェクト」と呼称する。

アニメーションやビデオゲームといった創作コンテンツ上で、エネルギーの塊が強く発光、形状変化するといった現象はアクションや格闘を主題としたコンテンツ内ではよく見かけ、攻撃方法の1つや特殊効果として広く用いられている。1960年代の特撮映像作品において既に光線の表現方法として確認できる。キャラクターの体内から溢れ出る超常的な力を対戦相手に目掛けて放出する事で攻撃している様子を映像化し、そうした表現が後に漫画やアニメーションといったコンテンツへ影響を及ぼしたものと見られている。近年の日本国内において、このような創作表現は愛好家の間で「エネルギー波」と呼ばれるようになった。ビデオゲームにおいては、攻撃における特殊効果全般をエフェクトという用語で表現し、エフェクトの代表例としてエネルギー波表現が含まれている。国外においても、アニメーションやCGの中でのVFXとして、エネルギー波表現は一般的な表現対象であると認識されている。その他にも、炎や雷・水といった自然現象を元にした創作的な攻撃描写も総じてエフェクトと呼ばれるが、本研究では漫画やアニメーション作品で多用されているエネルギー波表現に焦点を当てて取り扱う。本研究では、慣例に倣いこれらの特殊視覚効果をエネルギー波と呼称し、「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。エネルギーとは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。

3次元のビデオゲーム内において、エネルギー波を表現する為に用いる技術で現在一般的なものは、テクスチャ表現である。1枚の矩形ポリゴンで表現するビルボードテクスチャでは、エネル

ギー波を単一の 2 次元画像として事前に複数枚用意し、プレイヤーの視点変更などに応じて画像を回転・切り替える事で、1つのエネルギー波を表現している。また、球体や円柱形などのプリミティブ形状に対してテクスチャを貼り付けることでエネルギー波を表現する場合もある。テクスチャ表現はデータ容量が非常に軽く、リアルタイム描画性に優れているが、事前に用意したテクスチャ画像のみで表現を行うため、3次元空間中のエネルギー分布状態を考慮していない。そのため、プレイヤー視点を動的に任意に変更した場合、その視点から見たエネルギー波の光の強さや形状を正確に表現することは不可能となる。

ポリゴンなどの境界表現以外で、3次元空間中のエネルギー分布状態を表現するデータ構造で代表的なものはボクセルを用いたボリュームデータ表現である。ボリュームデータとは任意の3次元空間を一定の領域でサンプリングした集合データであり、個々をボクセルと呼称する。ボリュームデータは物体の内部構造や濃淡、煙等の不定形自然現象を表現する場合に適当なデータ構造である。ボクセル毎にシミュレート結果を事前計算しておき形状全体を表現する。必要に応じて事前計算結果を呼び出して使用する事が基本である。高精細な表示が可能であるが、その分データ容量が肥大化する傾向に有り、数値積分など計算コストの高い表示処理を求められる。近年では GPU を用いた高速な表示手法が数多く研究されているが、インタラクティブなコンテンツ内等で使用するには、データ容量や使用状況に制限がかかる事が多い。

ゲーム等の創作コンテンツではしばしば、炎や雷、水といった自然現象を模してエフェクトを作成する。こういった不定形自然現象に対してユーザー制御を目的とした研究は多岐に渡る。しかし近年では物理挙動ベースのシミュレーションが多く、コンテンツ内でエフェクト形状の再計算を伴う表現を想定した場合は計算コストが高い傾向にあり、ゲーム内では事前計算した結果を再生する利用方法が主流である。

本研究ではビデオゲームにおけるエネルギー波表現に着目し、空間中の稠密なエネルギー分布状態を高速にレンダリングする手法構築を目的とする。ゲームエフェクトの既存表現技術では、主に空間中に粒子の集合体を配置し、粒子位置にエネルギー波を構成するアニメーションテクスチャを複数パーツ組み合わせる事でエネルギー波表現を行っている。事前に手の込んだアニメーションテクスチャやパーティクルの運動を設定し、ゲームプレイ中に作成済みのデータを再生す

る方法が主流となる。その為、作成済みエフェクトに対するインタラクション性は低い傾向にある。本研究では、球体や螺旋といった形状関数を用いてエネルギーの分布状態を基底する事で、ゲーム制作工程における事前素材作成無しにエネルギー波描画を行う。関数を用いてエネルギー波形状を画面の描画処理毎に計算し直す事で、パラメータやカメラ視点変更といったユーザーからのインタラクションに対応した出力結果をリアルタイムに更新可能となる。

本研究では基礎的な形状に特化した解析的積分手法と、曲線的な形状に対応可能な数値的積分手法を提案する。球体等の単純形状を示す関数を被積分関数とした場合、原始関数が初等関数の組み合わせで表現可能となる為、積分区間が自明であれば積分値が解析的に求まる。曲線といった複雑な形状を示す関数が被積分関数の場合、原始関数を求める事が困難になる為、数値積分を用いてエネルギー分布状態を計算する。こうしたエネルギー波の分布状態を表す関数を予め基底し、カメラから見た関数値を描画処理毎に計算し直す。カメラ視点がどの位置にあってもエネルギー分布が連続的に存在する為、空間的に隙間のない稠密なエネルギー波分布を描画可能となる。各積分手法には得手不得手な点が存在するが、両手法とも既存のテクスチャ表現やパーティクル表現に比べ、エフェクトの見た目に関するテクスチャデータ作成やアニメーション設定といった事前作業が不要となる。ボリュームデータ表現も分布状態の動的変化には不向きであるという点から、空間中の稠密なエネルギー分布状態を動的に調節可能な本提案手法はエネルギー波表現に適している。

以上の目的を達成する提案手法を開発し、各種描画結果より本手法の有用性を検証した。

目次

第 1 章	序論	1
1.1	ゲームエフェクトについて	2
1.2	エネルギー波について	3
1.3	ゲームにおけるエフェクト作成について	5
1.4	本研究の目的	6
1.5	論文構成	7
第 2 章	関連研究	9
2.1	リアルタイムグラフィックスにおけるエネルギー波表現	10
2.2	関数を用いた形状表現に関する研究	11
2.3	関数を用いたエフェクト表現に関する研究	12
第 3 章	エネルギー波生成手法	15
3.1	本章の概要	16
3.2	エネルギー波生成手法の基本方針	18
3.2.1	関数分布によるエネルギー波形状の基底	18
3.2.2	積分区間の決定	18
3.2.3	積分区間中のエネルギー値を算出	19
3.2.4	エネルギー波の描画	20
3.3	解析的積分手法	21
3.3.1	球体分布関数と円柱分布関数の基底	21
3.3.2	解析的手法における積分区間の決定	22
3.3.3	解析的積分によるエネルギー値算出	22
3.3.4	形状変形と移動制御	25
3.3.5	隠面処理	26
3.4	数値的積分手法	27
3.4.1	一般陽関数曲線を用いたエネルギー波分布関数の基底	27
3.4.2	媒介変数同士の対応付け	28
3.4.3	一般陽関数を用いた被積分関数	29
3.4.4	数値的積分によるエネルギー値算出	30

3.4.5	色彩決定	32
3.4.6	移動表現	33
3.4.7	提案手法の適用例	33
3.4.7.1	三角関数による螺旋形状	34
3.4.8	三角関数和による関数形状	35
3.4.9	n 次ベジェ曲線形状	35
3.4.10	B-Spline 曲線形状	36
第 4 章	検証	37
4.1	本章の概要	38
4.2	検証環境	38
4.3	描画結果	38
4.3.1	関数分布による各種形状の描画結果	39
4.3.2	パラメータ変更による描画結果	39
4.3.3	色彩変更による描画結果	39
4.3.4	移動表現による描画結果	42
4.3.5	隠面処理による描画結果	44
4.4	実行速度	45
4.5	表現例の比較	49
4.6	課題	50
第 5 章	結論	57
5.1	研究のまとめと成果	58
5.2	現状の課題	59
5.3	今後の展開	60
	謝辞	61
	参考文献	63
	発表業績	69

目次

1.1	創作コンテンツ内のエネルギー波表現.	4
1.2	本研究における目標エネルギー波形状	5
1.3	本論文の章構成.	8
3.1	レイキャスティング法の模式図	16
3.2	球関数における各視線に応じたエネルギー量変化の様子.	18
3.3	積分区間と描画面の位置関係.	19
3.4	CPU と GPU の処理の流れ比較.	20
3.5	端点膨張型のエネルギー波形状模式図.	22
3.6	積分区間制御によるカプセル形状表現.	24
3.7	各画素のデプス値による積分区間の変更	27
3.8	t,s の対応付を表す模式図	28
3.9	各視線に応じたエネルギー量変化の様子	30
3.10	螺旋を覆うバウンディングボックスと視線との交点	31
3.11	円柱と視線の交差による分点数削減	32
4.1	球体と円柱による描画結果. $rgba = (0.8, 0.8, 0.2, 1.0)$ を基準色とし, 算出したエネルギー量を乗算して色彩を決定している.	40
4.2	各曲線の描画結果. $rgba = (0.5, 0.5, 1.0, 1.0)$ を基準色とし, 算出したエネルギー量を乗算して色彩を決定している.	41
4.3	カメラ位置を変更した出力結果比較.	42
4.4	調整係数 α を変更した出力結果比較.	42
4.5	調整係数 γ を変更した出力結果比較.	43
4.6	ベジェ曲線の制御点位置を変更した出力結果比較.	43
4.7	B-Spline 曲線の制御点位置を変更した出力結果比較.	44
4.8	基準色に応じた色変化の様子.	44
4.9	異なる基準色曲線の計算結果合算の様子.	45
4.10	減衰式を適用した曲線表示例. (a),(d) の $j = 0.0$ は減衰無しとなる. 減衰のパラメータ j 以外は (a) ~ (f) まで全て同値である.	46

4.11 減衰式を適用した螺旋表示例. (a),(d) の $j = 0.0$ は減衰無しとなる. 減衰のパラメータ j 以外は (a) ~ (f) まで全て同値である.	47
4.12 時間変化に応じた中心位置変更によるエネルギー波の伸長表現.	48
4.13 時間変化に応じた積分区間変更によるエネルギー波の伸長表現.	48
4.14 隠面処理の有無による表示比較. 左が隠面処理無し, 右が隠面処理有りとなる.	53
4.15 遠方カメラ配置 - 分点削減有無による螺旋曲線比較	54
4.16 遠方カメラ配置 - 分点削減有無による B-Spline 曲線比較	54
4.17 近辺カメラ位置 - 分点削減有無による結果比較	55
4.18 アニメコンテンツおよび本手法との作例 - 端点膨張型直線.	55
4.19 アニメコンテンツおよび本手法との作例 - 螺旋曲線. 中心の直線はベジェ曲線を用いて追加.	56
4.20 アニメコンテンツおよび本手法との作例 - 制御点による曲線.	56

表 目 次

4.1	検証用 PC の環境	38
4.2	各積分手法における描画要素	39
4.3	解析手法 - 実行速度 (FPS)	49
4.4	数値手法 - 実行速度 (FPS)	50
4.5	分点数削減を伴う実行速度 (FPS)	51

第 1 章

序論

1.1 ゲームエフェクトについて

近年、アニメーションや映画、ビデオゲームといった創作コンテンツ上で3次元コンピュータグラフィックス(以下3DCG)を用いた様々な表現が開発・利用されている。人体の写実的表現 [1][2] や光学効果による場面演出 [3], 流体物質の運動 [4][5] 等多岐に渡り3DCG技術が活用されている。特に映画やゲームといったコンテンツ上では、エフェクトと呼ばれる視覚効果が用いられている。こうした視覚効果は Visual Effects(VFX) と呼ばれ、コンテンツ内の状況やシーンを彩る手段として発展してきた。映画とゲームはどちらも描画したいシーンを構築した後、実際の映像として描画を行う。映画とゲームではコンテンツの性質上の違いがあり、映画では映像を視聴者へ届ける一方向的なメディアであり、シーンの描画に数分から数時間、あるいは数日かける事が可能である。対してゲームはユーザー側からのインタラクションによって描画すべき映像が変化する双方向的なメディアである。その為、シーンの描画には非常に高速な、1秒間に数十回の映像描画が必要となる。前者の映像描画に多量の時間をかけることができる方針をプリレンダリング、後者の素早く映像描画を行う必要がある方針をリアルタイムレンダリングと呼び、映像制作においては技術体系や考え方を区別して捉えている。

ゲームにおけるエフェクトには、ゲーム内の出来事をユーザーへ伝達する際の伝達効率を向上する説明的役割、シーン内の出来事を装飾し盛り上げる演出的役割が有り、総じてゲームコンテンツの魅力を向上させる役割である [6][7]。説明的役割例として、キャラクターが有利な状態へ移行した際に特別な光を発する、ゲーム内UI上で選択しているボタンを強調する等が挙げられる。演出的役割例として、キャラクターが敵に目掛けて魔法を照射する、攻撃が当たったら大きな爆発を付加する、場面全体にカメラの効果を加える等が挙げられる。ゲームエフェクトは大別して2種類ある。ポストプロセス・エフェクトとパーティクル・エフェクトである。ポストプロセス・エフェクトは、画面全体に対して視覚効果を付与する手段である。シーン全体に霧をかけ神妙な雰囲気演出したり、光の当たり方をシミュレートし昼間から夕方にかけての時間変化を一瞬に切り替える演出を行うなどの利用方法がある。パーティクル・エフェクトは、キャラクターや建物といったオブジェクトと同様にゲーム内シーン中に存在する物(オブジェクト)としての視覚効果である。炎や水・雷といった自然現象、魔法や光線といった架空現象等を、そのシーン中に存在

するオブジェクトとして表示する。近年において視覚効果オブジェクトとしてのエフェクトを作成するツールは、パーティクルという粒子を基本単位とする事が多い為、俗にパーティクル・エフェクトと呼ばれている。多量のパーティクルの集合体で流体的なエフェクトそのものを表現する利用方法や、パーティクルの位置座標にアニメーションテクスチャを配置する等、エフェクト素材の発生位置としての利用方法がある。ただし、現在においても矩形ポリゴンや球体といったプリミティブ形状を用いてエフェクトを配置する技法も一般的である。その為、本研究ではこのシーン中に存在する視覚的効果オブジェクトの事を単に「エフェクト」と呼称する。

1.2 エネルギー波について

アニメーションやビデオゲームといった創作コンテンツ上で、エネルギーの塊が強く発光、形状変化するといった現象はアクションや格闘を主題としたコンテンツ内ではよく見かけ、攻撃方法の1つや特殊効果として広く用いられている。1960年代の特撮映像作品 [8][9] において既に光線の表現方法として確認できる。キャラクターの体内から溢れ出る超常的な力を対戦相手に目掛けて放出する事で攻撃している様子を映像化し、そうした表現が後に漫画やアニメーションといったコンテンツへ影響を及ぼしたものと見られている。近年の日本国内において、このような創作表現は愛好家の間で「エネルギー波」と呼ばれるようになった。ビデオゲームにおいては、攻撃における特殊効果全般をエフェクトという用語で表現し、エフェクトの代表例としてエネルギー波表現が含まれている。国外においても、アニメーションやCGの中でのVFXとして、エネルギー波表現は一般的な表現対象であると認識されている。その他にも、炎や雷・水といった自然現象を元にした創作的な攻撃描写も総じてエフェクトと呼ばれるが、本研究では漫画やアニメーション作品で多用されているエネルギー波表現に焦点を当てて取り扱う。本研究では、慣例に倣いこれらの特殊視覚効果をエネルギー波と呼称し、「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。エネルギーとは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。

エネルギー波表現は現実世界のレーザー光線や、炎・雷といった物理現象を攻撃方法へ転じた

ものが発想の発端だと考えられる。その反面、「エネルギー」や「波」という単語が用いられているが、あくまで創作コンテンツ内での架空現象であり、現実世界の物理特性とは異なる挙動表現がなされている。作品毎に程度の差は存在するが、ある程度の共通項が見受けられる。

- 掌や口といったキャラクター身体あるいは武器といった発射元から勢い良く放出される。
- 放出後の移動速度は作品内においてある程度高速であるが、目視可能な範囲内に留まる。
- 他物体および他エネルギー波との衝突が生じ、衝突に応じてエネルギー波形状が大きく変化あるいは消失する。衝突された物体も多くの場合破壊表現が伴う。
- エネルギー波の形状および移動軌跡は直線・直進のみならず、曲線的な要素を含む。

以上の様に、実世界における光学現象とは異なる特性を持つ。

従来のキャラクター同士の身体による直接的な近接戦闘表現から、エネルギー波表現を飛び道具に見立てての間接的な戦闘表現が可能となった。特に、飛び道具でありながらもエネルギー波同士の衝突によって発射元に荷重がかかる表現がなされ、エネルギー波を介して鏝迫り合いが生じる。拮抗状態を描いた緊迫感のあるシーンが表現可能となったり、戦闘が決着する際の決め技として多用され、エネルギー波表現の人気および知名度の向上に繋がっている。

これらの視覚効果に対し、「エネルギー波」という呼称が最初に用いられた作品は、漫画「ドラゴンボール」の作品中においてである [10]。作品中では、体内エネルギーを光線状に放出する攻撃を「エネルギー波」、弾状に放出する攻撃を「エネルギー弾」と呼称した。作品は世界中で人気となり、その認知度から他の創作コンテンツにおいても同様の現象が「エネルギー波」と呼ばれることが多くなった。図 1.1 は「ドラゴンボール」のアニメーション作品におけるエネルギー波の一例である。

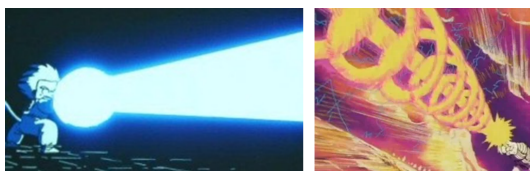


図 1.1: 創作コンテンツ内のエネルギー波表現。

出典 “ドラゴンボール” © バードスタジオ/集英社・フジテレビ・東映アニメーション

エネルギー波は当初直線的に放出される表現が多く見られたが、現在では曲線的な特徴を伴った表現も多数確認できる。エネルギー波表現における曲線的な特徴は、エネルギー波が攻撃対象方向に向かって移動する際の曲線的な軌跡、螺旋といった周期的な曲線によるエネルギー波の形状特徴などが挙げられる。曲線的な形状や軌跡は「ドラゴンボール」における「連続エネルギー弾」[11]や、「葬送のフリーレン」における「ゾルトラーク」[12]、螺旋といった周期的な形状特徴は「超電磁ロボ コン・バトラー V」における「超電磁スピン」[13]や、「機動武闘伝 G ガンダム」における「ビームリボン」[14]の様に、様々な作品において表現が行われている。こうした既存コンテンツ表現例を参考にし、本研究の目標対象形状は大きく分けて4点とする。

- 一直線上に目標へ向かって伸びていく直線形状。
- 端点に膨らみを持つ直線形状。
- 曲線的な経路で目標点へ向かう曲線形状。
- 螺旋に代表される周期的な形状特徴を持つ曲線形状。

図 1.2 は各エネルギー波形状の模式図である。

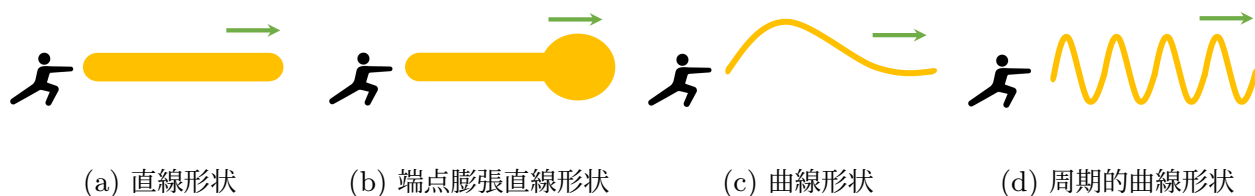


図 1.2: 本研究における目標エネルギー波形状

1.3 ゲームにおけるエフェクト作成について

ビデオゲームにおけるエフェクトはゲーム場面を盛り上げるための装飾として、ビデオゲーム登場初期から用いられている。一般的に、操作するキャラクターの絵や背景画とは別にエフェクト専用の画像を用意し、ゲーム中のタイミングに合わせてエフェクトを表示する。エフェクトデザイナーは予めエフェクトの形状や色、発生地点やタイミング、動きや模様の変化といったアニメーションを連番画像として作成する。キャラクターのモーションと組み合わせてエフェク

トを表示することで、ゲーム画面により一層の臨場感を付加できる。一般的な Digital Content Creation(DCC) ツールでは、エフェクトの一連の動きを 1 つのファイルに連番画像としてまとめたデータを flipbook やアニメーションテクスチャなどと呼称している (本稿では以下アニメーションテクスチャと表記する)。矩形ポリゴン上にアニメーションテクスチャを投影する手法が最も一般的なエフェクト描画手法である。1 枚の矩形に表示する場合や、複数枚の矩形ポリゴンを組み合わせて、複雑なエフェクト形状の表示をしている場合もある。円柱形や円錐形といった単純形状にアニメーションテクスチャを投影することで、更に多様なエフェクト表現を可能とする。

近年ではゲームエンジンがエフェクト作成に特化した機能を提供している。Unity では VFX Graph[15], UE4 では Niagara[16] が VFX 生成システムとして提供されている。これらのツールではパーティクル・エフェクト作成に念頭を置いた機能提供がなされている。また、パーティクルにアニメーションテクスチャを貼り付ける事で、様々な表現を可能としている。アニメや漫画のエフェクト表現には炎・雷・水・風といった自然現象をモチーフにした表現が多数存在している。ビデオゲームにおいてパーティクルエフェクトは従来のプリミティブ形状に投影するアニメーションテクスチャ表現に比べアニメ、これらの漫画的表現に向けた手法である。

近年のハードウェアは大量のパーティクルを管理可能なスペックを備えているが、空間中に稠密なエネルギー分布状態を基底しつつ、インタラクションに応じたデータ変化に対応するには、事前準備や計算領域の限定といった制約が未だ残る。また、ゲームにおいてエフェクトは装飾的な役割を担うことが主であるため、ゲーム起動中にエフェクト表示工程に割り当てる計算時間やメモリリソースは限定されることが多い。エフェクト描画に係る処理速度やデータ量が少量な状態下においても表現力豊かな手法の存在は望ましい。

1.4 本研究の目的

本研究ではビデオゲームにおけるエネルギー波表現に着目し、空間中の稠密なエネルギー分布状態を高速にレンダリングする手法構築を目的とする。ゲームエフェクトの既存表現技術では、主に空間中に粒子の集合体を配置し、粒子位置にエネルギー波を構成するアニメーションテクスチャを複数パーツ組み合わせる事でエネルギー波表現を行っている。事前に手の込んだアニメー

ションテクスチャやパーティクルの運動を設定し、ゲームプレイ中に作成済みのデータを再生する方法が主流となる。その為、作成済みエフェクトに対するインタラクション性は低い傾向にある。本研究では、球体や螺旋といった形状関数を用いてエネルギーの分布状態を基底する事で、ゲーム制作工程における事前素材作成無しにエネルギー波描画を行う。関数を用いてエネルギー波形状を画面の描画処理毎に計算し直す事で、パラメータやカメラ視点変更といったユーザーからのインタラクションに対応した出力結果をリアルタイムに更新可能となる。

本研究では基礎的な形状に特化した解析的積分手法と、曲線的な形状に対応可能な数値的積分手法を提案する。球体等の単純形状を示す関数を被積分関数とした場合、原始関数が初等関数の組み合わせで表現可能となる為、積分区間が自明であれば積分値が解析的に求まる。曲線といった複雑な形状を示す関数が被積分関数の場合、原始関数を求める事が困難になる為、数値積分を用いてエネルギー分布状態を計算する。こうしたエネルギー波の分布状態を表す関数を予め基底し、カメラから見た関数値を描画処理毎に計算し直す。カメラ視点がどの位置にあってもエネルギー分布が連続的に存在する為、空間的に隙間のない稠密なエネルギー波分布を描画可能となる。各積分手法には得手不得手な点が存在するが、両手法とも既存のテクスチャ表現やパーティクル表現に比べ、エフェクトの見た目に関するテクスチャデータ作成やアニメーション設定といった事前作業が不要となる。ボリュームデータ表現も分布状態の動的変化には不向きであるという点から、空間中の稠密なエネルギー分布状態を動的に調節可能な本提案手法はエネルギー波表現に適している。

以上の目的を達成する提案手法を開発し、各種描画結果より本手法の有用性を検証した。

1.5 論文構成

本論文は全5章からなる。2章では、本研究における関連研究と先行研究を紹介すると共に、本研究との位置付けを明確にする。3章では関数分布を用いたエネルギー波表現手法について提案する。4章では提案手法を用いたエネルギー波表現について検証を行う。最後に、5章にて本論文の結論をまとめ、今後の展望を述べる。図 1.3 に各章の関連を示す。

3章では2つの手法を提案する。単純形状に適した解析的手法と、曲線形状に適した数値的

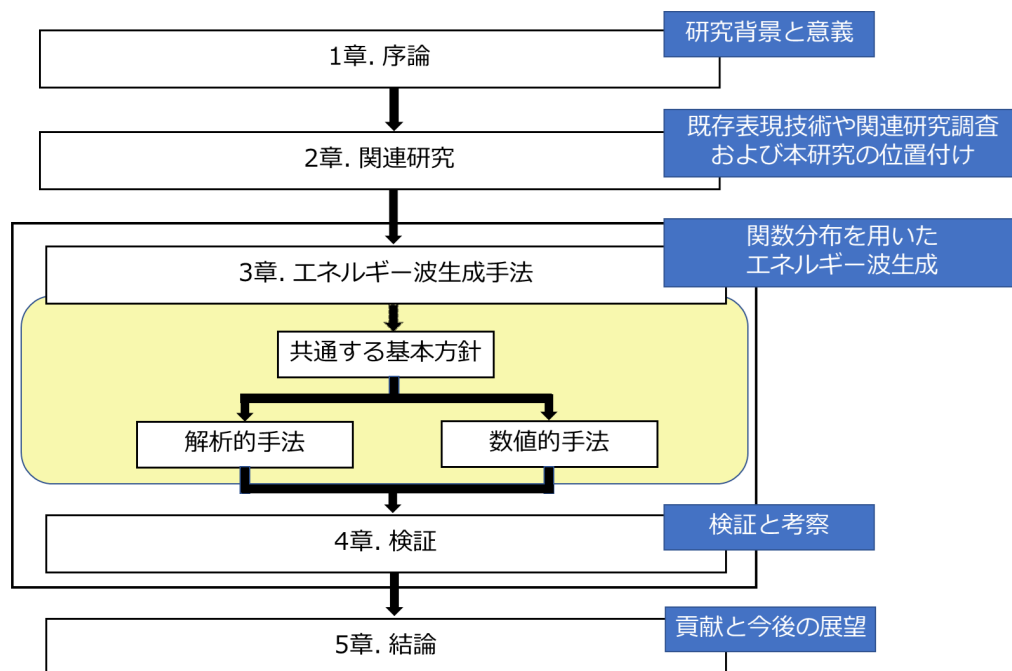


図 1.3: 本論文の章構成.

法である。初めに両手法に共通している提案手法全体の処理工程を提示し、続いて各手法毎の詳細について提示していく。4章では両手法について検証を行う。

本論文では、エネルギー波をレンダリングして2次元画像を得る事を「エネルギー波の描画」、エネルギー波の形状・色合い・移動といった、描画対象の意匠性を指す場合を「エネルギー波の表現」と記載する。

第 2 章

関連研究

本章では、本研究における関連研究と先行研究を紹介すると共に、本研究との位置付けを明確にする。

2.1 リアルタイムグラフィックスにおけるエネルギー波表現

3次元のビデオゲーム内において、エネルギー波を表現する為に用いる技術で現在一般的なものは、テクスチャ表現である。1枚の矩形ポリゴンで表現するビルボードテクスチャ [17] では、エネルギー波を単一の2次元画像として事前に複数枚用意し、プレイヤーの視点変更などに応じて画像を回転・切り替える事で、1つのエネルギー波を表現している。球体や円柱形などのプリミティブ形状に対してテクスチャを貼り付けることでエネルギー波を表現する場合もある。テクスチャ表現はデータ容量が非常に軽く、リアルタイム描画性に優れているが、事前に用意したテクスチャ画像のみで表現を行うため、3次元空間中のエネルギー分布状態を考慮していない。そのため、プレイヤー視点を動的に任意に変更した場合、その視点から見たエネルギー波の光の強さや形状を正確に表現することは不可能となる。計算機やゲーム機種が多様になった現代においても、計算能力が高くないデバイス上でのゲーム内エフェクト表現として欠かせない技術となっている。

高性能なPCゲームや最新コンシューマ機において、パーティクル・エフェクトも広く使用されているエフェクト表現である。1.3節にて述べたVFX GraphやNiagaraではパーティクルの集合体で不定形現象を表現したり、アニメーションテクスチャの表示位置にパーティクルの位置を利用する。ゲームエンジン以外にも、エフェクト作成専門ツールとしてEffekseer [18] 等があり、エフェクト作成環境は入手し易くなっている。

ポリゴンなどの境界表現以外で、3次元空間中のエネルギー分布状態を表現するデータ構造で代表的なものはボクセルを用いたボリュームデータ表現 [19] である。ボリュームデータとは任意の3次元空間を一定の領域でサンプリングした集合データであり、個々をボクセルと呼称する。ボリュームデータは物体の内部構造 [20] や濃淡、煙等の不定形自然現象 [21] を表現する場合に適切なデータ構造である。2次元のテクスチャと同様に、ボクセルの解像度を高く設定すれば高精細な表示が可能であるが、その分データ容量が肥大化する傾向に有り、数値積分など計算コストの高い表示処理を求められる [22]。近年ではGraphics Processing Unit(GPU)を用いた高速な

表示手法 [23][24][25] が数多く研究されているが、インタラクティブなコンテンツ内等で使用するには、データ容量や使用状況に制限がかかる事が多い。解像度が固定化されてしまう、データ量が肥大化傾向にある、形状をゲーム内で動的に変更する事が困難である等の理由から、ゲームエフェクトに利用する場合は制約も多い方針となる。ポリューメトリックな流体用エフェクト作成ツールに EmberGen[26] がある。ゲームでの利用を想定した炎や煙、爆発の流体的運動を高速に計算し、アニメーションテクスチャを生成可能である。エネルギー波の持つコンテンツ上の特性と炎の特性は異なる為、本ツールを単に用いただけではエネルギー波生成は困難である。

ゲーム等の創作コンテンツではしばしば、炎や雷、水といった自然現象を模してエフェクトを作成する。こういった不定形自然現象に対してユーザー制御を目的とした研究 [27][28] は多岐に渡る。しかし近年では物理挙動ベースのシミュレーションが多く、コンテンツ内でエフェクト形状の再計算を伴う表現を想定した場合は計算コストが高い傾向にあり、ゲーム内では事前計算した結果を再生する利用方法が主流である。

Nowrouzezahrai ら [29] はフォトンマッピングアルゴリズムを利用した意匠操作が可能な光表現を開発した。しかしこの手法も映画等の高精細なアニメーション利用を前提としており、ビデオゲームへの単純な適応は困難である。仁藤ら [30] はエネルギー波が衝突する際の飛沫形状について分類を行い、制御点を用いて調整可能な手法を開発した。リアルタイムに処理が可能なエネルギー波表現ではあるが、焦点は衝突時の変形処理であり、エネルギー波の形状構成はパーティクルの集合としている。本研究の目的に照らし合わせた場合、パーティクル・エフェクトと同様の問題点を抱える。

2.2 関数を用いた形状表現に関する研究

空間中の濃度分布表現の代表的手法には陰関数を用いた手法が挙げられる。Blinn の BlobbyModel[31] や西村らによる Metaball[32] がその代表例である。陰関数表現では関数の値を元に、空間を物体の内側、外側、表面と分類する。等値面を物体の表面として定義する手法であり、有機的な曲面形状や幾何学的形状を容易に表現可能である。また、陰関数の組み合わせによる分布状態の定義は、表面の混合、変形といった位相変化を伴う処理が堅牢かつ容易であるた

め多用されている。複数の陰関数を組み合わせることで、水流や飛沫等の流体表現 [33] や、人体や動物といった形状を表現可能である [34][35]。しかし表示物体が持つ透明度や濃度値は表示に反映されることはなく、対象の表面を表示する事が目的 [36] である。複数変数の連続関数による FRep オブジェクト [37] を使用した表現では形状の他に、形状同士の加算や減算といったオペレータや、形状同士の関係性も関数表現を用いて行う。FRep オブジェクトを元にオペレータや関係性に独自性をもたせ、濃度値をオブジェクト表面に反映させることで透過面や濃淡といった表現が可能である [38][39][40]。これらの手法は形状表面の描画に主眼を置いたものであり、エネルギー波形状の様な空間中の稠密な濃度分布を描画する目的には直接的には不適當である。

2.3 関数を用いたエフェクト表現に関する研究

渡辺 [41] は関数分布によるエネルギー波表現を行った。距離関数によるスカラー場を基底し、ピクセルに映るスカラー値の合計を算出する事で、エネルギー波形状を表現している。トーラス型形状や 2 次ベジェ曲線による自由曲線形式での形状表現を可能とした。特に曲線形式においては、部位毎にエネルギー量の強弱を調整可能であり、不均一な形状表現を実現すると共に曲線上での移流アニメーションも可能となった。その反面エネルギー分布を求める際に近似解でエネルギー分布を求めている。数値積分を行うにあたり、積分区間中の分割数と描画結果はトレードオフな関係となる。一般的な数値解析における積分計算の問題と同様に、近似計算の回数に応じて処理速度低下が見られる。2 次ベジェ曲線を大量に繋ぎ合わせれば多数の形状種類を表現可能であるが、曲線の個数が増加すれば全体の処理速度が著しく低下する。渡辺は形状表現事例として螺旋を示しているが、多数の曲線を繋ぎ合わせて螺旋を描画した結果、リアルタイムな表示は不可能であった。その為、大域的に 1 つの形状を表現可能な関数種類が増えることが望ましい。

本研究はビデオゲームにおけるエネルギー波表現に着目し、空間中の稠密なエネルギー分布状態を高速にレンダリングする手法構築を目的とする。既存エフェクト表示技術では任意視点からの光の強さの正確性に欠けていたり、アニメーションテクスチャの配置関係によっては非連続的な描画結果となってしまう。エネルギー波形状を構成する空間中の稠密なエネルギー分布状態を関数化する事で、描画処理毎に視点位置に応じたエネルギー波の光の強さや形状特性を計算し、

描画結果を得る。これまでリアルタイム表示が不可能であった分布関数のうち、三角関数曲線やパラメトリック曲線を目標形状に含める事で、先行研究からの表現発展を目指す。関数曲線は創作コンテンツ内でエネルギー波形状やエネルギーの光跡として用いられる事の多い形状種であり、リアルタイム描画可能となる事での有意性が高い。本研究ではビデオゲームにおけるエネルギー波表現に着目し、空間中の稠密なエネルギー分布状態を高速にレンダリングする手法構築を目的とする。ゲームエフェクトの既存表現技術では、主に空間中に粒子の集合体を配置し、粒子位置にエネルギー波を構成するアニメーションテクスチャを複数パーツ組み合わせる事でエネルギー波表現を行っている。事前に手の込んだアニメーションテクスチャやパーティクルの運動を設定し、ゲームプレイ中に作成済みのデータを再生する方法が主流となる。その為、作成済みエフェクトに対するインタラクション性は低い傾向にある。本研究では、球体や螺旋といった形状関数を用いてエネルギーの分布状態を基底する事で、ゲーム制作工程における事前素材作成無しにエネルギー波描画を行う。関数を用いてエネルギー波形状を画面の描画処理毎に計算し直す事で、パラメータやカメラ視点変更といったユーザーからのインタラクションに対応した出力結果をリアルタイムに更新可能となる。

本研究では基礎的な形状に特化した解析的積分手法と、曲線的な形状に対応可能な数値的積分手法を提案する。球体等の単純形状を示す関数を被積分関数とした場合、原始関数が初等関数の組み合わせで表現可能となる為、積分区間が自明であれば積分値が解析的に求まる。曲線といった複雑な形状を示す関数が被積分関数の場合、原始関数を求める事が困難になる為、数値積分を用いてエネルギー分布状態を計算する。こうしたエネルギー波の分布状態を表す関数を予め基底し、カメラから見た関数値を描画処理毎に計算し直す。カメラ視点がどの位置にあってもエネルギー分布が連続的に存在する為、空間的に隙間のない稠密なエネルギー波分布を描画可能となる。各積分手法には得手不得手な点が存在するが、両手法とも既存のテクスチャ表現やパーティクル表現に比べ、エフェクトの見た目に関するテクスチャデータ作成やアニメーション設定といった事前作業が不要となる。ボリュームデータ表現も分布状態の動的変化には不向きであるという点から、空間中の稠密なエネルギー分布状態を動的に調節可能な本提案手法はエネルギー波表現に適している。

以上の目的を達成する提案手法を開発し，各種描画結果より本手法の有用性を検証した．

第 3 章

エネルギー波生成手法

3.1 本章の概要

本章では本研究におけるエネルギー波表現手法について述べる。提案手法の基本的な考え方は3次元空間中にエネルギー分布関数を基底し、その上で視点から描画面中の各画素に対して視線を飛ばし、各画素を通る直線上のエネルギー量を線積分によって求めるというものである。ボリュームレンダリングの一種であるレイキャスティング法 [42] と基本的な概念は同様である。ボリュームレンダリングとは、3次元空間中における物理量や内部構造を持つオブジェクト等の物理量分布データを描画目的とした技術の総称である。2次元のピクセルデータに対して、3次元的な描画対象のデータをボリュームデータと呼ぶ。レイキャスティング法は視点の位置から描画面に対して視線を飛ばし、その視線方向に沿ってボリュームデータ値を積算していく手法である。描画面のピクセル毎に同様の処理を繰り返し行う事で、最終的な描画結果を得る。図 3.1 はレイキャスティング法における視線と描画面、ボリュームデータの関係を図示したものである。

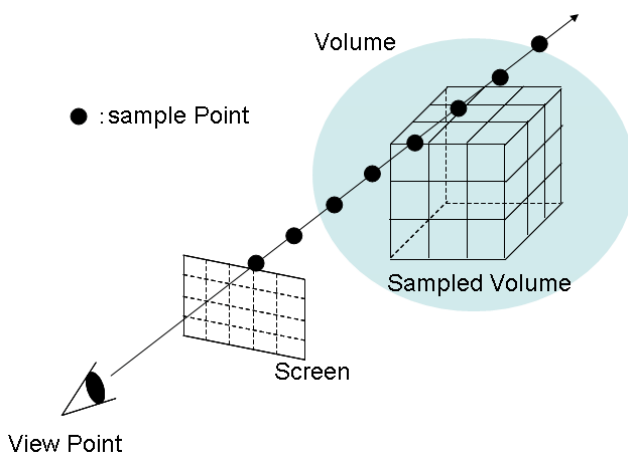


図 3.1: レイキャスティング法の模式図

ボリュームレンダリングは描画対象となるボリュームデータの事前用意が必要であるが、提案手法ではこれを関数化し描画処理毎にその都度計算を行う。関数化による利点として、ゲーム制作工程における事前用意データが不要であったり、描画対象に対するインタラクション表現に向いている。場に対してエネルギー分布関数を基底し、スカラー場に対する線積分と見なす事でピクセル毎のエネルギー値を算出していく。提案手法では3次元分布状況の近似値であるボリュームデータを生成することなく、視線に沿ったエネルギーの分布関数を線積分することで最終的な

描画結果を得る。

提案手法では線積分計算がピクセル毎に独立しており、かつ同じ関数式を用いる為並列処理に適している。本手法では OpenGL のシェーダー機能で並列計算を行う。提案手法の手順は以下である。

1. 3次元空間中にエネルギーの分布状況を関数を用いて基底
2. 積分区間の決定
3. 積分区間中のエネルギー値を算出
4. エネルギー波の描画

以降、3.2 節では初めに、本提案手法の基本方針となる全体像について述べる。描画対象のエネルギー波について、空間中にエネルギーが分布している状態を関数によって基底する。描画面の各画素に対して視線を設定し、視線上における関数値を積分計算する事でエネルギー波を描画する。本研究は表現対象のエネルギー波形状種類に応じて異なる分布関数を提案する。また、提案分布関数の特性に応じて 2 種類の積分方法に大別する。

3.3 節では表現対象形状が球体および円柱の分布関数における解析的な積分方針について述べる。線積分に用いる被積分関数の原始関数が、初等関数の組み合わせで表現可能な場合、解析的積分が可能となる。解析的積分では計算処理が非常に高速であり、エネルギーの分布状態を最も正しい値として算出可能となる。本研究では、任意の地点を基準とした点分布、任意の線分を基準とした線分布を解析的積分手法の対象とする。

3.4 節では表現対象形状が曲線形状の分布関数における数値的な積分方針について述べる。線積分に用いる被積分関数の原始関数が算出困難である場合、数値的積分を用いてエネルギー値計算を行う。数値的積分では近似値を求めることになる為、計算精度と計算時間はトレードオフな関係となるが、解析的積分に比べ描画可能なエネルギー波形状種類が増加できるといった特徴を持つ。本研究では、三角関数曲線やパラメトリック曲線に一定の条件を定めた曲線を一般陽関数曲線と呼称し、一般陽関数曲線を数値的積分手法の対象とする。

3.2 エネルギー波生成手法の基本方針

本手法では方針の異なる 2 つの積分手法を提案する。被積分関数によって積分方針が異なることになるが、エネルギー波描画に至る大筋の工程は両積分手法とも同様である。その為、本節では共通項となる本提案手法の全体像を述べる。

3.2.1 関数分布によるエネルギー波形状の基底

エネルギー波形状はコンテンツ作品中において、強い発光を伴いながら描かれる事が多い。その為本手法ではエネルギー波形状の中心となる点や線上が最もエネルギー値が強く、基準位置から離れるにつれエネルギー値が下がる分布形式とする。3次元空間中の任意地点 \mathbf{P} とし、 \mathbf{P} におけるエネルギー値 $F(\mathbf{P})$ とすると、本手法におけるエネルギー波分布関数は式 (3.1) を基本とする。

$$F(\mathbf{P}) = \frac{\alpha}{E(\mathbf{P})}. \quad (3.1)$$

この時 $E(\mathbf{P})$ は表現対象のエネルギー波形状に合わせて変化する。 α はエネルギー波の光の強さを管理する動的な調整係数とする。図 3.2 は球体を例に、エネルギー分布に対して視線毎にエネルギー値が異なる様子を示す。

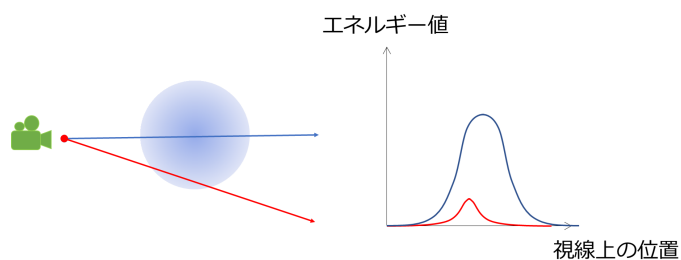


図 3.2: 球関数における各視線に応じたエネルギー量変化の様子。

3.2.2 積分区間の決定

視点の位置ベクトルを \mathbf{V}_s , 視点から描画面に対して視線を飛ばし、視点からみて十分遠い距離にある視線上の点を \mathbf{V}_g とする。 \mathbf{V}_s と \mathbf{V}_g の範囲内が積分区間となる。図 3.3 は積分区間と描

画面の位置関係を表している。同様にして、描画面の各画素に対して視線および \mathbf{V}_g を設定する。

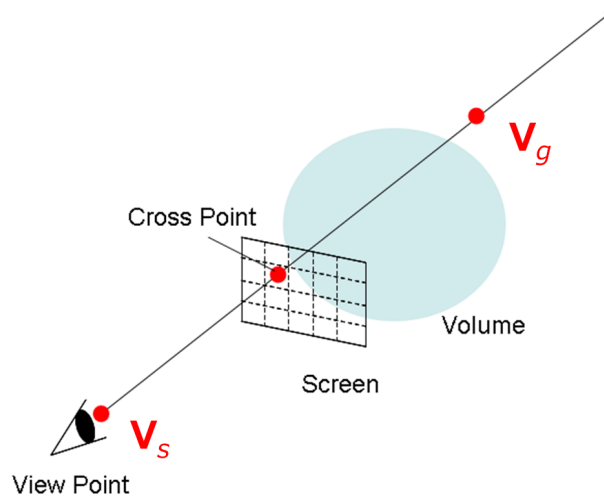


図 3.3: 積分区間と描画面の位置関係。

\mathbf{V}_s は各画素で変化はせず，共通の値を用いる。

3.2.3 積分区間中のエネルギー値を算出

基本方針は，スカラー場に対する線積分と見なす事が出来る．初めに線積分の経路に該当する視線ベクトルを，媒介変数 t で表現する． \mathbf{V}_s ， \mathbf{V}_g の 2 点間を結ぶ直線を GS とし，直線 GS 上の任意地点 \mathbf{V} を媒介変数 t を用いて式 (3.3) で表す。

$$\mathbf{V}(t) = (\mathbf{V}_g - \mathbf{V}_s)t + \mathbf{V}_s. \quad (3.2)$$

ただし $0 \leq t \leq 1$.

直線 GS が線積分経路， $\mathbf{V}(t)$ におけるスカラー値を経路区間中で積分する事で，1 画素におけるエネルギー値を算出する。

エネルギー波形状を表す関数に $\mathbf{V}(t)$ を代入し， t について積分する事で対象画素のエネルギー値が求まる．式 (3.3) がエネルギー値算出用の積分式となる。

$$\int_0^1 F(\mathbf{V}(t))dt. \quad (3.3)$$

3.2.4 エネルギー波の描画

基本方針では線積分計算が画素毎に独立しており、かつ同じ関数式を用いる為並列処理に適している。本手法では OpenGL のシェーダー機能で並列計算を行う。OpenGL の画像処理工程の中で最終工程に当たるフラグメントシェーダー内で積分計算を行う。3DCG では空間中に配置したオブジェクトの位置やカメラ設定、光源位置などから最終的に 2 次元のディスプレイに映る画像をシェーダー内で計算して描画している。シェーダーは GPU で計算を行う為、CPU と比べ分岐処理や再帰処理は不得意であるが、単純なデータを 1 度に大量に扱う並列処理は得意である。本手法において、エネルギー波の投影面を生成する処理は、1 画素につき式 (3.3) を 1 度計算することになる。この計算を画素毎に行う事で、最終的な描画面全体を得る。本手法では複数の画素値を 1 度に並列計算することで、全体としての処理速度を向上させている。図 3.4 は描画結果の計算処理において、CPU 処理及び、GPU 処理の流れを比較した模式図である。

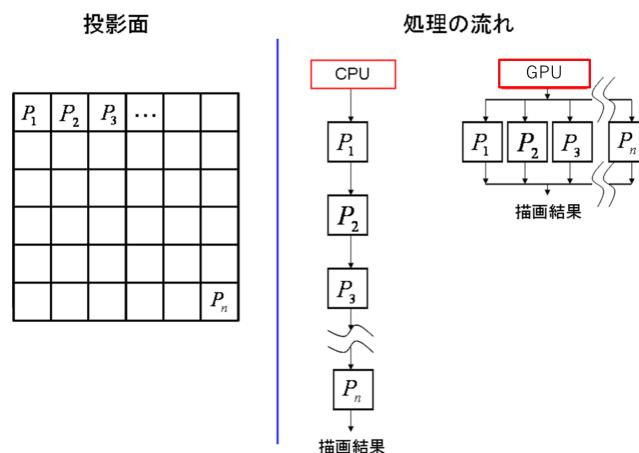


図 3.4: CPU と GPU の処理の流れ比較。

最後に、これまでの工程で算出した画素毎の計算結果を用いて色彩を決定する。表現を行いたい色彩を本研究では基準色と呼称し、基準色の値 (r, g, b, a) をそれぞれ 0.0 ~ 1.0 の範囲で既定しておく。積分の計算結果はスカラー値となっている為、基準色の各成分に対して計算結果を乗算する事で最終的なピクセルの色値とする。

3.3 解析的積分手法

次に具体的な積分手法のうち、解析的積分手法について述べる。表現を行いたいエネルギー波目標形状のうち、直線状や球状といった構造がシンプルな形状が対象となる。

3.3.1 球体分布関数と円柱分布関数の基底

表現目標である直線形状および端点が膨張した様な直線形状について、それぞれの構成要素である球体と円柱について関数分布を基底していく。直線形状は大部分を円柱形状として表現する。円柱形状の両端点に対して半球形状を加えることで有限な長さの線分として描画可能となる。この円柱形状の両端点に半球形状を加えたカプセル形状を、本手法の直線形状とする。

任意の球体の中心座標を \mathbf{M} 、 \mathbf{M} を起点に放射状にエネルギーが広がる分布関数を $S(\mathbf{P})$ とし、以下の式 (3.4) で球を表す関数式とする。 $S(\mathbf{P})$ は球体の中心地点ほどエネルギー値が高く、中心地点から距離が離れるにつれ低くなり、任意のパラメータ α_s によってエネルギーの度合いを操作できる。

$$S(\mathbf{P}) = \frac{\alpha_s}{|\mathbf{P} - \mathbf{M}|} \quad (3.4)$$

任意の直線を基準の中心線とし、その直線から広がる分布関数を $C(\mathbf{P})$ とする。 $C(\mathbf{P})$ は円柱の中心線でのエネルギー値が高くなり中心線から距離が離れるにつれ低くなる関数であり、任意のパラメータ α_c によってエネルギーの度合いを操作できる。円柱の中心線が延びる任意の方向ベクトルを \mathbf{D} とし、 \mathbf{D} は単位ベクトルとする。

$$C(\mathbf{P}) = \frac{\alpha_c}{\sqrt{|\mathbf{P} - \mathbf{M}|^2 - ((\mathbf{P} - \mathbf{M}) \cdot \mathbf{D})^2}} \quad (3.5)$$

カプセル形状については、領域に応じて球体関数と円柱関数を切り替える事で表現する。点 \mathbf{M} を含み \mathbf{D} を法線とする平面を I_1 、もう片方の端点位置としての任意座標 \mathbf{M}_2 を含み $-\mathbf{D}$ を法線とする平面を I_2 とする。2つの平面において領域を区分し、平面間の領域では円柱関数である式 (3.5) を、逆向きの領域では球体関数である式 (3.4) を用いる事でカプセル形状を表現する。各平面において球体用パラメータ α_s と円柱用パラメータ α_c の値を一致させることで、全ての空間中

でエネルギー分布が連続になる。

エネルギー波は一直線上に目標地点へ向けて移動する際、発射位置や先頭部分といった端を他の箇所よりも肥大化して表現する事が多い。図 3.5 は、球体分布関数および円柱分布関数を組み合わせた、本解析的積分手法の総合的な目標形状を示す模式図である。

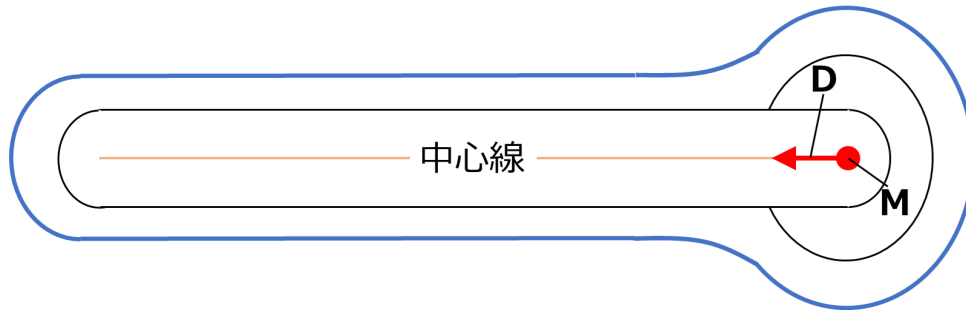


図 3.5: 端点膨張型のエネルギー波形状模式図.

3.3.2 解析的手法における積分区間の決定

球体形状について、3.2 節にて述べた通り、積分の開始点を視点位置、各画素へ向けた視線にある視点位置から十分遠い地点を積分の終了点とする。カプセル形状について、開始地点および終了地点は球体形状のみの場合と同一だが、区間中で平面 I を境目とし球体と円柱の分布関数を切り替える。詳細は次項にて述べる。

3.3.3 解析的積分によるエネルギー値算出

$\mathbf{V}_g = \mathbf{E}$, $\mathbf{V}_s = \mathbf{S}$ とし、 $\mathbf{E} - \mathbf{S}$ を \mathbf{Q} とする。エネルギー値を算出したい地点は $\mathbf{V}(t)$ 上になる為、 $\mathbf{P} = \mathbf{V}(t)$ とし、 $\mathbf{V}(t)$ の各成分を式 (3.4), 式 (3.5) に代入する。

$$S(t) = \frac{\alpha_s}{\sqrt{U(t)}} \quad (3.6)$$

$$U(t) = |\mathbf{V}(t) - \mathbf{M}|^2$$

$$C(t) = \frac{\alpha_c}{\sqrt{W(t)}} \quad (3.7)$$

$$W(t) = U(t) - ((\mathbf{V}(t) - \mathbf{M}) \cdot \mathbf{D})^2$$

式 (3.6), 式 (3.7) のうち, $U(t)$, $W(t)$ は t の 2 次式といえる. これを $U(t) = At^2 + 2Bt + C$, $W(t) = Dt^2 + 2Et + F$ とすると, $U(t)$, $W(t)$ は常に正の値をとることが保証できる事から, $S(t)$, $C(t)$ の線積分式はそれぞれ式 (3.8), 式 (3.9) で求まる.

$$\int_0^1 S(t)dt = \left[\frac{\alpha_s \sinh^{-1} \left(\frac{2(tA + B)}{4AC - (2B)^2} \right)}{\sqrt{A}} \right]_0^1 \quad (3.8)$$

$$\begin{aligned} & \text{ただし } A \neq 0 \\ & 4AC - (2B)^2 \neq 0 \end{aligned}$$

$$\int_0^1 C(t)dt = \left[\frac{\alpha_c \sinh^{-1} \left(\frac{2(tD + E)}{4DF - (2E)^2} \right)}{\sqrt{D}} \right]_0^1 \quad (3.9)$$

$$\begin{aligned} & \text{ただし } D \neq 0 \\ & 4DF - (2E)^2 \neq 0 \end{aligned}$$

本解析手法で用いるカプセル形状のエネルギー分布は, 領域区分用平面を境目とした式 (3.4) と式 (3.5) の合計値である. 式 (3.4) を用いる区間と式 (3.5) を用いる区間の調整は, 線分 GS と平面の交点を利用する. GS と平面 I_1 の交点における媒介変数 t_1 の値を以下の式 (3.10) で求める.

$$t = \frac{(\mathbf{D} \cdot \mathbf{M}) - (\mathbf{D} \cdot \mathbf{S})}{\mathbf{D} \cdot \mathbf{Q}} \quad (3.10)$$

もう片方の平面 I_2 との交点における媒介変数 t_2 も同様にして求める。各 t を境目としたカプセル形状を示すエネルギー分布関数は以下の式 (3.11) とする。視線の開始位置や方向に応じて、区間は適宜変化する。

$$\int_0^{t_1} S_1(t)dt + \int_{t_1}^{t_2} C(t)dt + \int_{t_2}^1 S_2(t)dt \quad (3.11)$$

図 3.6 は積分式を平面 I_1, I_2 で切り替えることで表現したカプセル形状の模式図である。

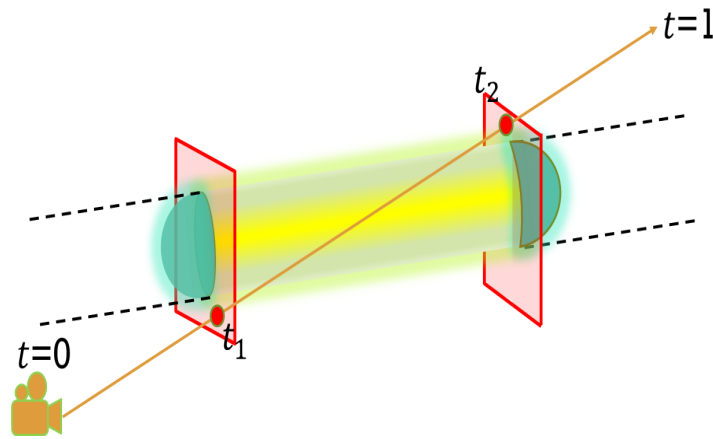


図 3.6: 積分区間制御によるカプセル形状表現。

ベクトル成分については以下のものとする。

$$\mathbf{S} = (s_x, s_y, s_z)$$

$$\mathbf{M} = (m_x, m_y, m_z)$$

$$\mathbf{D} = (d_x, d_y, d_z)$$

$$\mathbf{Q} = (q_x, q_y, q_z)$$

式 (3.8) の A, B, C はそれぞれ以下のように求まる。

$$A = q_x^2 + q_y^2 + q_z^2$$

$$B = q_x s_x + q_y s_y + q_z s_z - q_x m_x - q_y m_y - q_z m_z$$

$$C = (s_x - m_x)^2 + (s_y - m_y)^2 + (s_z - m_z)^2$$

式 (3.9) の D , E , F はそれぞれ以下のように求まる.

$$D = (1 - d_x^2)q_x^2 + (1 - d_y^2)q_y^2 + (1 - d_z^2)q_z^2 - 2(d_x d_y q_x q_y + d_x d_z q_x q_z + d_y d_z q_y q_z)$$

$$\begin{aligned} E = & m_x d_x d_x q_x + m_x d_x d_y q_y + m_x d_x d_z q_z \\ & + m_y d_x d_y q_x + m_y d_y d_y q_y + m_y d_y d_z q_z \\ & + m_z d_x d_z q_x + m_z d_y d_z q_y + m_z d_z d_z q_z \\ & - d_x d_x q_x s_x - d_x d_y q_y s_x - d_x d_z q_z s_x \\ & - d_x d_y q_x s_y - d_y d_y q_y s_y - d_y d_z q_z s_y \\ & - d_x d_z q_x s_z - d_y d_z q_y s_z - d_z d_z q_z s_z \\ & - m_x q_x - m_y q_y - m_z q_z \\ & + q_x s_x + q_y s_y + q_z s_z \end{aligned}$$

$$\begin{aligned} F = & 2(m_x d_x d_x s_x + m_x d_x d_y s_y + m_x d_x d_z s_z \\ & + m_y d_x d_y s_x + m_y d_y d_y s_y + m_y d_y d_z s_z \\ & + m_z d_x d_z s_x + m_z d_y d_z s_y + m_z d_z d_z s_z \\ & - d_x d_y s_x s_y - d_y d_z s_y s_z - d_x d_z s_x s_z \\ & - m_x m_y d_x d_y - m_y m_z d_y d_z - m_x m_z d_x d_z \\ & - m_x s_x - m_y s_y - m_z s_z) \\ & + m_x^2 + m_y^2 + m_z^2 + s_x^2 + s_y^2 + s_z^2 \\ & - m_x^2 d_x^2 - m_y^2 d_y^2 - m_z^2 d_z^2 - d_x^2 s_x^2 - d_y^2 s_y^2 - d_z^2 s_z^2 \end{aligned}$$

描画面を構成する画素毎に対して視線を飛ばし、以上の手順によるエネルギーの密度分布計算を行うことで、全体の描画結果を得る.

3.3.4 形状変形と移動制御

エネルギー波は形状変化を伴いながら目標地点に向かって移動をする. 解析的手法ではエネルギーの分布状況や任意の位置ベクトルを変更することで、エネルギー波の変形と移動を表現する.

球体部分、カプセル部分それぞれの移動は、任意の点 \mathbf{M} の位置を変更する事で行う. 本手法では球体とカプセルに与える任意位置ベクトル \mathbf{M} を同一に設定しているが、個別に設定することも可能である.

エネルギー波の形状変形は、球体における任意のパラメータ α_s 、カプセルにおける任意のパラメータ α_c をそれぞれ変更する事で表現する. カプセルは伸びる方向ベクトル \mathbf{D} も任意であるた

め、 \mathbf{D} を変更する事でも変形表現が可能である。パラメータ α_s を増加させる事で、 \mathbf{M} を中心とする球体の大きさを大きく、パラメータ α_c を増加させる事で、カプセルの太さを太くする事が可能である。パラメータ α_s, α_c を動的に変更することで、エネルギー波形状の動的な制御を行う。

3.3.5 隠面処理

エネルギー波に隠面処理を施すことで、一般的なオブジェクトとの前後関係を表現し、エネルギー波表現としての有用性を高める。3DCG 空間をレンダリングする場合、投影面の各画素は色値と同時に、その色値が 3DCG 空間中において、カメラからどの程度離れているかという位置情報を持つ。この位置情報は一般的にデプス値と呼ばれる。3DCG では視点からは他の物体で影になっていたり、視線とは逆向きになっている面や線を隠面と呼び、隠面部分を描画しない事で、描画面上での立体感を高めている。視点から見て、デプス値よりも後ろの空間が隠面部分となり、敢えて描画しない区間となる。エネルギー波も一般的なオブジェクトと同様に、視点から見て隠面部分を描画しない事で、他のオブジェクトとの前後関係を表現する。

初めにエネルギー波以外で用意した全てのオブジェクトを 3DCG 空間中に配置し、この状態で各画素のデプス値を取得する。取得したデプス値を d とし、エネルギー波の隠面消去処理に適応するため、 d の値を 0 から 1 の範囲に正規化した値 g を以下の式 (3.12) で求める。

$$g = \frac{d}{|\mathbf{E} - \mathbf{S}|} \quad (3.12)$$

次に、各画素における積分区間の終了地点 \mathbf{E} を、式 (3.12) で求めた値 g を用いて変更する。変更した終了地点を \mathbf{E}' とし、以下の式 (3.13) のように表す。視点 \mathbf{S} からみて \mathbf{E}' よりも後ろの区間が、エネルギー波における隠面部分となる。

$$\mathbf{E}' = (\mathbf{E} - \mathbf{S})g + \mathbf{S} \quad (3.13)$$

最後に投影面を視点から視線方向へ一番近い位置に配置することで、エネルギー波と他オブジェクトの前後関係を考慮した表現を実現する。エネルギー波における隠面部分は他オブジェクトを配置した状態のデプス値から求めるため、他オブジェクトの個数に因らず処理は一定である。図 3.7 は各画素のデプス値によって積分区間を変更する状況の模式図である。

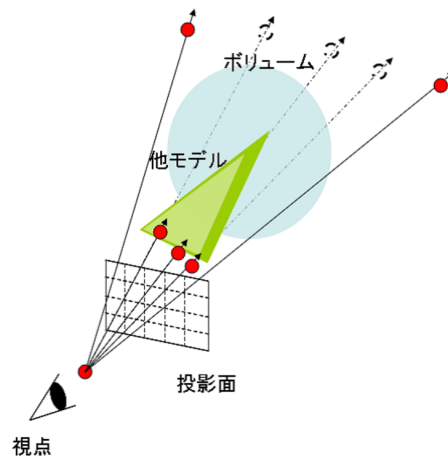


図 3.7: 各画素のデプス値による積分区間の変更

3.4 数値的積分手法

続いて、数値的積分手法について述べる。表現を行いたいエネルギー波形状目標形状のうち、曲線的な形状が対象となる。

3.4.1 一般陽関数曲線を用いたエネルギー波分布関数の基底

初めに、エネルギー波分布関数を定め媒介変数表記する。エネルギー波は攻撃対象目標へ向けて移動する様な表現が多用される為、曲線形状においてもエネルギー波の大域的な進行方向を特定軸と定め、提案分布関数は特定軸方向へは線形的に単調増加するものとする。特定軸を x 軸とした場合、曲線上の任意地点を \mathbf{C} とすると、成分を媒介変数 s を用いて式 (3.14) で表す。

$$\mathbf{C}(s) = \begin{pmatrix} \gamma s \\ y(s) \\ z(s) \end{pmatrix} \quad (3.14)$$

特定成分は調整係数 γ による線形的な単調増加、それ以外の成分を陽的な関数表記する上記式による曲線を、以降本論文では一般陽関数曲線と呼称する。

3.4.2 媒介変数同士の対応付け

本解析的手法では視線および曲線に用いているそれぞれの媒介変数 t, s を更に別の媒介変数 u を用いて対応付ける。 u の値に応じて、 t と s が一対の値を取る事となる。 この媒介変数同士の対応を取らない場合は 2 変数関数の重積分とみなせるが、 その場合線積分上の各点で曲線形状の全地点からのエネルギー値を計算することとなってしまう、 元々の曲線に沿った形状特徴が喪失してしまう。 その為、 線積分上の各点と曲線形状の各点を一対の組と見なした対応付けが必要となる。 本手法では曲線の大域的な進行方向を \mathbf{D} とし、 媒介変数 s を式 (3.15) で算出する。

$$s = \mathbf{V}(t) \cdot \mathbf{D} \quad (3.15)$$

$t = 0$ の場合 t_s, s_s とし、 $t = 1$ の場合 t_g, s_g とすると媒介変数 u を用いて式 (3.16) とする。

$$\begin{cases} u_t = t_s + (t_g - t_s)u \\ u_s = s_s + (s_g - s_s)u \end{cases} \quad (3.16)$$

線積分の対象が u によって定まる。 図 3.8 は媒介変数 t と s からなるエネルギー量分布および線積分の経路を表す模式図となる。 媒介変数同士の対応を取らず、 2 変数関数による重積分は曲線

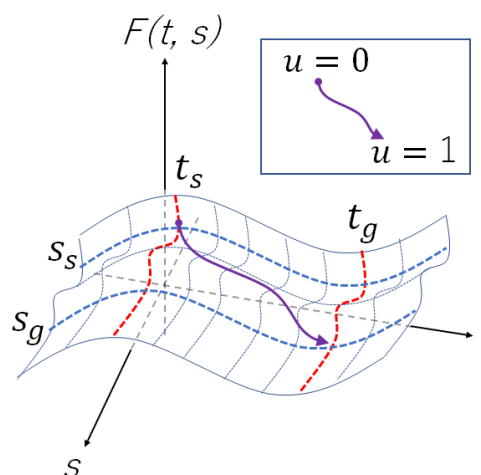


図 3.8: t, s の対応付を表す模式図

全体のエネルギー値を算出する事を指すが、 形状特徴が喪失してしまうという問題点がある。

3.4.3 一般陽関数を用いた被積分関数

一般陽関数を用いた数値的解析手法における提案被積分関数は $\mathbf{V}(u)$, $\mathbf{C}(u)$ の 2 点間距離に応じて減衰していくものとし, α を調整係数とした式 (3.17) として基底する.

$$F(t, s) = F(u) = \frac{\alpha}{|\mathbf{V}(u) - \mathbf{C}(u)|} \quad (3.17)$$

$\mathbf{V}_g - \mathbf{V}_s = \mathbf{V}_r$, $t_g - t_s = t_r$, $s_g - s_s = s_r$ とそれぞれ置き換え, $\mathbf{V}(u)$ と $\mathbf{C}(u)$ を以下のように表すものとする.

$$\begin{aligned} \mathbf{V}(u) &= \mathbf{V}_s + u_t \mathbf{V} \\ &= \mathbf{V}_s + t_s \mathbf{V}_r + u t_r \mathbf{V}_r \\ &= \begin{pmatrix} A + uB \\ C + uD \\ E + uF \end{pmatrix} \end{aligned} \quad (3.18)$$

$$\begin{aligned} \mathbf{C}(u) &= \begin{pmatrix} \gamma u_s \\ y(u_s) \\ z(u_s) \end{pmatrix} \\ &= \begin{pmatrix} \gamma s_s + \gamma s_r u \\ y(s_s) + y(s_r u) \\ z(s_s) + z(s_r u) \end{pmatrix} \\ &= \begin{pmatrix} I + Ju \\ K + y(Lu) \\ M + z(Nu) \end{pmatrix} \end{aligned} \quad (3.19)$$

さらに,

$$\begin{aligned} \beta &= ((A - I) + u(B - J))^2 \\ &\quad + ((C - K) + uD + y(Lu))^2 \\ &\quad + ((E - M) + uF + z(Nu))^2 \end{aligned} \quad (3.20)$$

とすることで,

$$|\mathbf{V}(u) - \mathbf{C}(u)| = \sqrt{\beta} \quad (3.21)$$

となる. この式 (3.21) を式 (3.17) に代入し, 一般陽関数の被積分関数とする. 式の展開中, $\mathbf{V}(u)$, $\mathbf{C}(u)$ の各成分のうち媒介変数 u に係らない項を $A = V_x^s + t_s V_x^r$, 係る項を $B = t_r V_x^r$ の様に代数的に置き換えている.

3.4.4 数値的積分によるエネルギー値算出

提案関数を積分する方法について述べる．1本の視線上のエネルギー量は，視線ベクトルの開始地点や終了地点，曲線の進行方向といったベクトル値より変化する．図 3.9 は一般陽関数を用いた螺旋曲線を例に，視線ベクトルに応じたエネルギー量について縦軸に $F(u)$ ，横軸に u を取りグラフ化したものである．左図が視線ベクトルと螺旋形状の位置関係を示し，右図がその視線ベクトルに応じたエネルギー量グラフを示す．

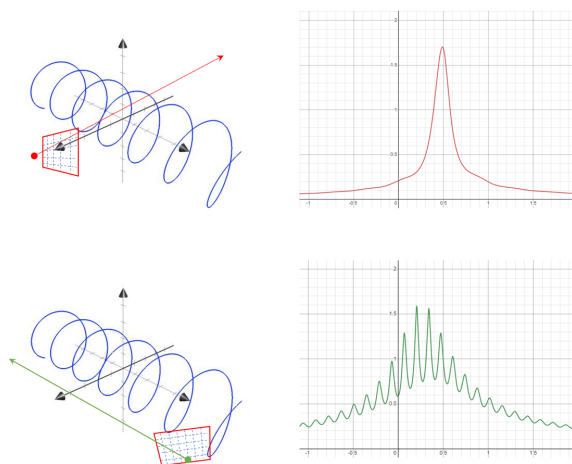


図 3.9: 各視線に応じたエネルギー量変化の様子

エネルギー波形状を表す分布関数種類によっては，積分する方向に応じて視線上のエネルギー関数，つまり線積分が表す関数形状が大きく変化する．後述する本手法の適用例である螺旋形状は特にその傾向が強い分布関数となっている．本提案関数を解析的に解くことは困難な為，近似解を求める事となる．近似解算出に際し，可能な限り積分計算の回数を減らしつつ高品質な描画結果を得られることが望ましい．近似解算出は主に，積分区間を一定法則で分割し，分割した小領域で近似計算を行い合計値を出す方針である．分割に用いる基点を以降は分点と呼称する．

平山 [43] によるテイラー展開を利用した積分法では，ある基準点において被積分関数を多項式で精度良く近似できる性質がある．しかし，本提案関数の様な周期的に変曲点が現れる多峰性関数に適応すると，本研究が求める積分範囲 ($0 \leq u \leq 1$) よりも狭い範囲で収束する傾向にあった為，少ない分点数で積分範囲全体を充足させるには不向きである．

本研究では渡辺 [41] による手法と同様に，ニュートン・コーツ公式の一種である合成シンプソン公式を用いて積分を行う．合成シンプソン公式は積分精度を高くする為には多くの分点が必要となり，処理速度と描画品質はトレードオフな関係になっている．

ボリュームレンダリングにおける高速化の方針として，描画結果に影響の少ないボクセルデータを削減するものがある [44][45][46]．不要部分を間引く発想を利用し，分布状態を関数化している本手法においても不要分点を間引く事で高速化を図る．自動積分法の適応型の考え [47] を参考にし，関数変化の緩急を積分計算よりも前の段階で，視線と対象曲線の位置関係から予め定める．通常の合成シンプソン公式では積分区間を等分割するが，本手法では視線と曲線の位置関係から予め定めた領域内外で異なる分割数を適用する．領域内では元々の分割数を保持し，領域外では分点数を間引くことで分割数を下げ，処理の高速化を図る．

対象曲線形状を包含するバウンディングボックスを設定し，バウンディングボックスと視線との交点を求める．本研究では値の範囲が振幅範囲内に限定できる周期関数の特性や，ベジェ曲線や B-Spline 曲線の凸包性を考慮し，バウンディングボックス形状を円柱とする．円柱の範囲を振幅や全制御点を包含可能な大きさとする事で，曲線形状全体を内包したバウンディングボックスが設定可能となる．交点よりも外側の領域はエネルギー波形状への影響は少ないとして，分点を減らして合成シンプソン公式を適用する．図 3.10 は視線と螺旋用バウンディングボックスとしての円柱の交点を示す模式図である．

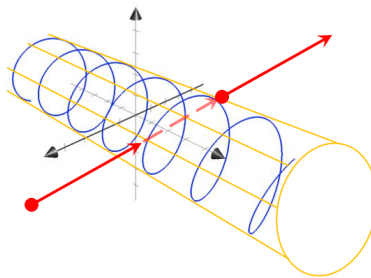


図 3.10: 螺旋を覆うバウンディングボックスと視線との交点

視線上の点 $V(u)$ と螺旋上の点 $C(u)$ は既に対応が取れている．この時螺旋の基準位置を C_b ，

螺旋の方向を \mathbf{C}_r とし、 $\mathbf{C}_r = (\mathbf{C}(u) \cdot (1.0, 0.0, 0.0), 0.0, 0.0)$ とする。 $\mathbf{V}(u), \mathbf{C}_r$ 間の距離が r となる u が交点位置となる為、 u についての式 (3.22) を解く。

$$\begin{aligned} r + \epsilon &= |(\mathbf{V}_s + u_t \mathbf{V}_r) - (\mathbf{C}_b + u_s \mathbf{C}_r)| \\ &= |(t_r \mathbf{V}_r - s_r \mathbf{C}_r) u + (\mathbf{V}_s + t_s \mathbf{V}_r) - (\mathbf{C}_b + s_s \mathbf{C}_r)| \end{aligned} \quad (3.22)$$

ϵ はオフセット領域調整用の数値である。 u についての 2 次式になるので、 解の公式より u が求まる。 2 つの実数解を $u_1, u_2 (u_1 < u_2)$ とすると u_1 が積分開始地点に近い交点、 u_2 が奥側の交点となる。 積分区間全体を都合 3 つの領域に分割し、 円柱を貫通している領域の分点数を m とすると、 円柱の外領域の分点数を $\frac{m}{n}$ (ただし $n < m$) と取る事で、 目的のエネルギー波形状を保ちつつ高速化を図る。 実数解が 1 以下の場合には円柱の領域外となるため、 視線全域において分点数を削減する。 図 3.11 は円柱の領域外の分点数を間引いている様子の模式図である。

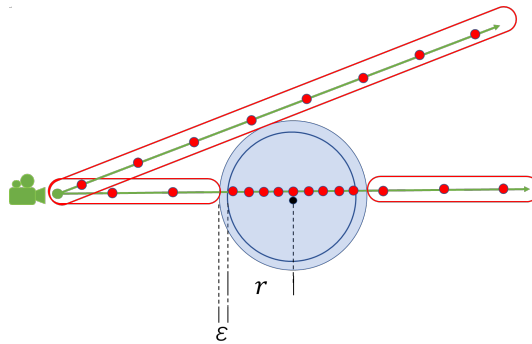


図 3.11: 円柱と視線の交差による分点数削減

3.4.5 色彩決定

本手法は式 (3.17) の結果を元にピクセル毎の色彩を決定する。 本手法ではシェーダーを使用してピクセル毎にエネルギー波分布関数を並列計算し、 計算結果を求める。 基準色の値 (r, g, b, a) をそれぞれ 0.0 ~ 1.0 の範囲で既定値として定めておく。 それぞれの成分に対して式 (3.17) の結果を乗算する事で最終的なピクセルの色値とする。

次に、 視点位置とエネルギー波の位置関係により、 エネルギー波形状が肥大化した様な描画結果になる事への補正方法について述べる。 視点位置によっては曲線の輪郭周辺のエネルギー量が増加し、 結果として輪郭を覆う様に霧がかかる領域が大きくなる変化である。 こうした出力状態に

ついて、エネルギー波の輪郭がはっきりした表現を好む場合もある。距離に応じたエネルギー値減衰補正を加える事で、選択可能な表現の幅を広げる。エネルギー波は創作コンテンツ上の架空現象ではあるが、現実の光の特性を利用し距離に応じて減衰させることで、輪郭を出しやすくする。本手法では線積分のパラメータを u_t を用いて、カメラ位置から遠方の地点にあるエネルギー値ほど値を減衰する。現実の光は距離の 2 乗分照度が低下するが、本研究では j 乗 (ただし j は正の実数) として減衰効果を制御可能なものとする。式 (3.23) は本研究における距離に応じた減衰式である。

$$F(u) = \left(\frac{\alpha}{|\mathbf{V}(u) - \mathbf{C}(u)|} \right) (1.0 - u_t)^j \quad (3.23)$$

3.4.6 移動表現

本手法では媒介変数 u の積分区間を $[0,1]$ とする事でエネルギー波形状全体を描画する。発射点から目標地点へ向けて放出されるエネルギー波の移動表現を例とした場合、一般陽関数曲線の媒介変数 u_s を区間 $[0,k]$ ($0.0 < k \leq 1.0$) とすることで、移動途中の形状を描画可能となる。積分区間の下端を 0.0、積分区間上端を k とした区間 $[0,k]$ の積分と、区間 $[k,1]$ の積分は式 (3.24) となる。

$$F(u) = \int_0^k \frac{\alpha}{|\mathbf{V}(u) - \mathbf{C}(u)|} du + \int_k^1 \frac{\alpha}{|\mathbf{V}(u) - \mathbf{C}(k)|} du \quad (3.24)$$

積分区間 k を時間変化に応じて制御する事で発射点から到着点へエネルギー波を放出する様を表現可能である。

3.4.7 提案手法の適用例

本章では前述した提案関数について、具体的に複数種類の曲線式を適用した際の式を示す。本論文では、

- 三角関数による螺旋曲線形状
- 三角関数和による周期的曲線形状
- n 次ベジエ曲線形状

- B-Spline 曲線形状

についての適用を例示する.

3.4.7.1 三角関数による螺旋形状

エネルギー波表現における代表的な三角関数表現に螺旋形状がある. エネルギー波をドリルの様な螺旋形状で放出することで敵を穿つさまを示したり, 竜巻の様なエネルギー波表現で良く用いられる. 螺旋の基準位置を \mathbf{C}_b , 螺旋上の任意地点を \mathbf{C} とすると, 媒介変数 s を用いて式 (3.25) で表す.

$$\mathbf{C}(s) = \begin{pmatrix} \gamma s \\ r \sin(s\theta) \\ r \cos(s\theta) \end{pmatrix} + \mathbf{C}_b \quad (3.25)$$

このとき, r は螺旋の半径を, γ は任意の調整係数とする.

$\mathbf{V}_g - \mathbf{V}_s = \mathbf{V}_r$, $t_g - t_s = t_r$, $s_g - s_s = s_r$ とそれぞれ置き換え, 式 (3.3)(3.16)(3.25) より $\mathbf{V}(u)$ と $\mathbf{C}(u)$ を以下のように表すものとする.

$$\begin{aligned} \mathbf{V}(u) &= \mathbf{V}_s + u_t \mathbf{V} \\ &= \mathbf{V}_s + t_s \mathbf{V}_r + ut_r \mathbf{V}_r \\ &= \begin{pmatrix} A + uB \\ C + uD \\ E + uF \end{pmatrix} \end{aligned} \quad (3.26)$$

$$\begin{aligned} \mathbf{C}(u) &= \mathbf{C}_b + \begin{pmatrix} \gamma u_s \\ r \sin(u_s\theta) \\ r \cos(u_s\theta) \end{pmatrix} \\ &= \begin{pmatrix} C_x^b + \gamma s_s + \gamma s_r u \\ C_y^b + r \sin(s_s\theta + s_r\theta u) \\ C_z^b + r \cos(s_s\theta + s_r\theta u) \end{pmatrix} \\ &= \begin{pmatrix} I + Ju \\ K + r \sin(G + uH) \\ L + r \cos(G + uH) \end{pmatrix} \end{aligned} \quad (3.27)$$

さらに,

$$\begin{aligned} \beta &= ((A - I) + u(B - J))^2 \\ &+ ((C - K) + uD + r \sin(G + uH))^2 \\ &+ ((E - L) + uF + r \cos(G + uH))^2 \end{aligned} \quad (3.28)$$

とすることで,

$$|\mathbf{V}(u) - \mathbf{C}(u)| = \sqrt{\beta} \quad (3.29)$$

となる. この式 (3.29) を式 (3.17) に代入し, 一般陽関数による螺旋関数とする.

3.4.8 三角関数和による関数形状

エネルギー波表現では, 幾何的に周期性のある形状や軌跡を伴う場合も多い. 周期的な曲線について, フーリエ級数展開で \sin 波と \cos 波に分解できる曲線であれば本手法で表現可能となる. ある一般陽関数曲線が三角関数和として表せた場合, 関数上の任意地点を \mathbf{C} とすると, 媒介変数 s を用いて式 (3.30) で表す.

$$\mathbf{C}(s) = \begin{pmatrix} \sum_n (a_n \sin(ns\theta) + b_n \cos(ns\theta)) \\ \sum_n (c_n \sin(ns\theta) + d_n \cos(ns\theta)) \end{pmatrix} \quad (3.30)$$

以降は式 (3.19) と同様に媒介変数 u に置き換え, 式 (3.17) となる様に u についてまとめる.

3.4.9 n 次ベジエ曲線形状

エネルギー波の放出軌跡は曲線になる場合も多い. 特にエネルギー波を連続で複数回放出するような飽和攻撃表現においては, 曲線軌跡を利用することで軌跡の重複を避け, 多数放出している様を表現している. 制御点を用いたパラメトリック曲線はこうした曲線形状や軌跡には有効な手段である. 既存研究 [41] では 2 次ベジエ曲線までの表現が限界だったのに対して, 本手法では 3 次以上のベジエ曲線 [48] にも適用可能である. ベジエ曲線は制御点列 $\{\mathbf{P}_0, \mathbf{P}_1, \dots, \mathbf{P}_n\}$ によって式 (3.32) で定まる.

$$B_i^n(s) = {}_n C_i s^i (1-s)^{n-i} \quad (3.31)$$

$$\mathbf{C}(s) = \sum_{i=0}^n B_i^n(s) \mathbf{P}_i \quad (0 \leq s \leq 1) \quad (3.32)$$

式 (3.14) の $y(s), z(s)$ はそれぞれ $\mathbf{C}(s)$ の y 成分と z 成分とする. 以降は式 (3.19) と同様に媒介変数 u に置き換え, 式 (3.17) となる様に u についてまとめる.

3.4.10 B-Spline 曲線形状

ベジェ曲線と同様に、一般に広く利用されている曲線として B-Spline 曲線 [48] がある。本手法は B-Spline 曲線にも適用が可能である。 $n + 1$ 個から成る制御点列 $\{\mathbf{P}_0, \mathbf{P}_1, \dots, \mathbf{P}_n\}$ と、位数 k およびノットベクトル $W = [x_0, x_1, \dots, x_{n+k}]$ を用いて以下の式 (3.33) により曲線形状が決定できる。

$$\mathbf{C}(s) = \sum_{i=0}^n N_k^i(s) \mathbf{P}_i \quad (2 \leq k \leq n + 1) \quad (3.33)$$

ここで、 $N_k^i(t)$ は B-Spline 基底関数を表し、以下のように定義される。

$$N_1^i(t) = \begin{cases} 1 & \text{if } x_i \leq t < x_{i+1} \\ 0 & \text{if } t < x_i \text{ or } t \geq x_{i+1} \end{cases}$$

$$N_k^i(t) = \frac{(t - x_i)N_{k-1}^i(t)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - t)N_{k-1}^{i+1}(t)}{x_{i+k} - x_{i+1}} \quad (3.34)$$

式 (3.14) の $y(s), z(s)$ はそれぞれ $\mathbf{C}(s)$ の y 成分と z 成分とする。以降は式 (3.19) と同様に媒介変数 u に置き換え、式 (3.17) となる様に u についてまとめる。

第 4 章

検証

4.1 本章の概要

本章ではこれまでの提案手法の検証結果について述べる。4.2 節では検証に用いた実行環境について述べる。4.3 節では本手法の描画結果を示し分析を行う。4.4 節では処理速度に関する検証結果について述べる。4.5 節ではコンテンツ利用時を想定した本手法による形状作例を示す。最後に4.6 節で本手法に対する議論について述べる。

4.2 検証環境

これまで述べた提案手法について、以下の表 4.1 のスペック PC にて描画および速度検証を行った。

表 4.1: 検証用 PC の環境

OS	Windows11 23H2
CPU	AMD Ryzen7 5600X, 3.70GHz
Memory	32GB
GPU	NVIDIA GeForce RTX 4060 Ti, 8GB
Programming	OpenGL 4.1, GLSL 4.1

本検証では WindowsOS 上で検証を行ったが、MacOS 上でも実行可能である事を確認している。

4.3 描画結果

本研究では表現対象形状種に応じて 2 つの積分手法を提示した。表 4.2 は 2 つの手法における出力描画に関する要素の内、共通した要素および固有の要素をまとめたものである。以降、それぞれの要素について結果を示す。ただし、数値手法中の分点削除については処理速度高速化の目的である為、4.4 節の処理速度と併せて描画結果を示す。

表 4.2: 各積分手法における描画要素

要素	解析手法 (球体・円柱)	数値手法 (一般陽関数曲線)
パラメータ	✓	✓
色彩 (減衰)	✓(×)	✓(✓)
移動	✓	✓
隠面	✓	×
分点	×	✓

4.3.1 関数分布による各種形状の描画結果

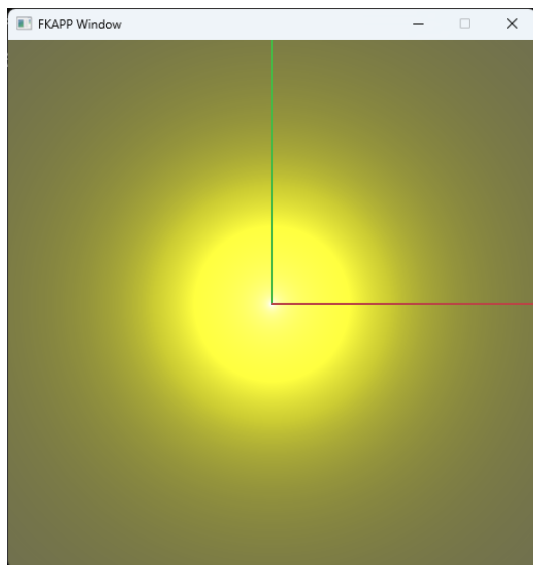
提案手法による描画結果について、球体および円柱分布を用いた解析手法を図 4.1 に、一般陽関数分布を用いた数値手法を図 4.2 に示す。

4.3.2 パラメータ変更による描画結果

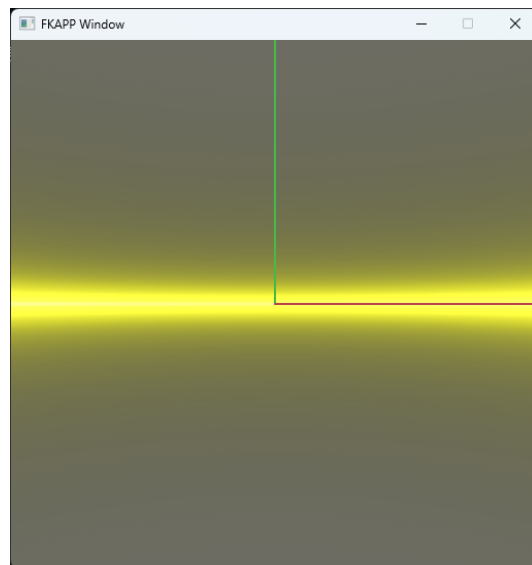
続いてプログラム中に調整可能な各機能について、動作の様子を示す。各種パラメータやカメラ位置、パラメトリック曲線における制御点位置は、プログラム実行中任意のタイミングで変更可能であり、プログラム実行速度への影響無く実時間実行可能である。図 4.3 はカメラ位置を変更した際の様子を示す。図 4.4 はエネルギー波の光の強さを管理している調整係数 α を変更した際の様子を示す。図 4.5 は螺旋の方向軸の伸長調整係数 γ を変更した際の様子を示す。図 4.6 はベジェ曲線の制御点を変更した際の様子を示す。図 4.7 は B-Spline 曲線の制御点を変更した際の様子を示す。

4.3.3 色彩変更による描画結果

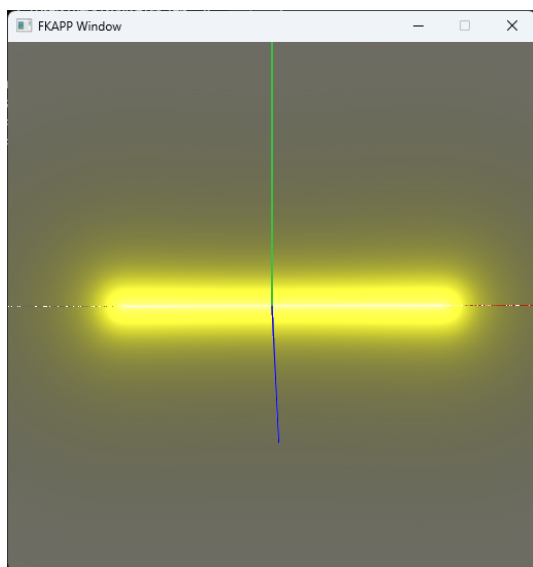
次に色彩表現に関する描画結果を示す。図 4.8 は異なる基準色を規定した場合の出力結果比較であり、エネルギー波分布関数の計算結果自体は同値である。図 4.9 は異なる基準色である上に凸の青色曲線と、下に凸の赤色曲線によって求まる色値を合算し最終的なピクセル値とした表示例である。基準色はプログラム実行中にも任意に変更可能であり、時間変化やエネルギー波周囲の環境変化に応じて好みに調整可能である。



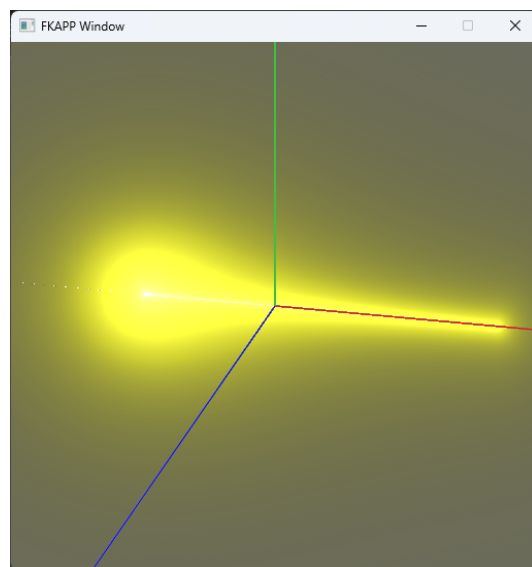
(a) 球体



(b) 円柱



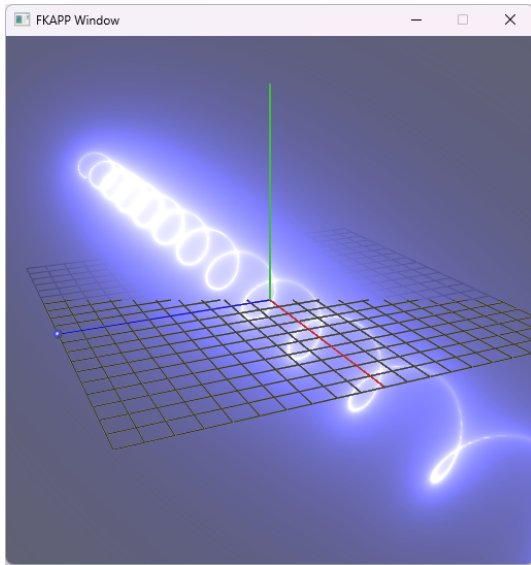
(c) カプセル



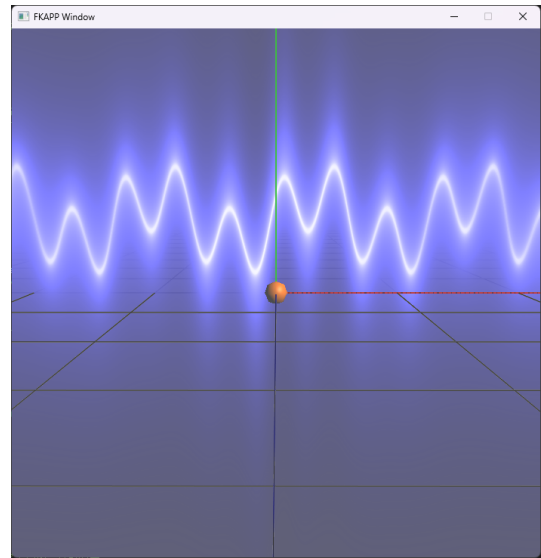
(d) 球体 + カプセル

図 4.1: 球体と円柱による描画結果. $rgba = (0.8, 0.8, 0.2, 1.0)$ を基準色とし, 算出したエネルギー量を乗算して色彩を決定している.

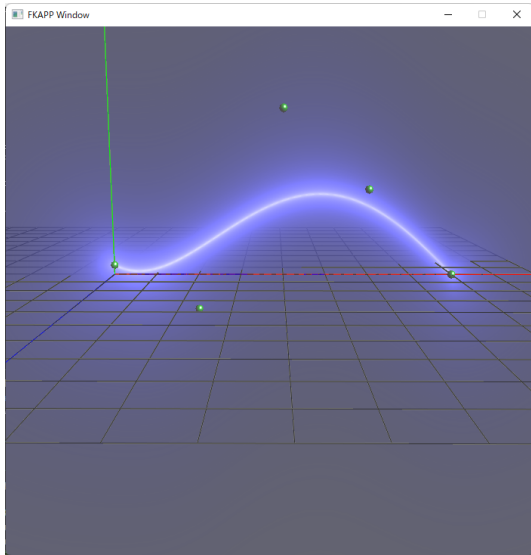
数値手法において, 視点位置とエネルギー波の位置関係により, エネルギー波形状が肥大化した様な描画結果になる. カメラ位置からエネルギー波分布までの距離に応じた減衰補正を加える事で, 極端な肥大化を解消する. 図 4.10 は減衰による色彩補正の有無の出力結果比較である. 補正を用いるか否かは最終的には表現者の判断となるが, 提案関数は補正処理に応じて見た目を変



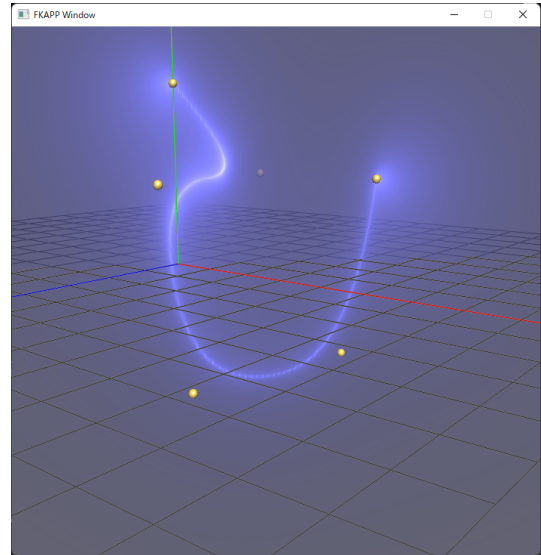
(a) 螺旋曲線



(b) 三角関数和曲線



(c) ベジエ曲線



(d) B-Spline 曲線

図 4.2: 各曲線の描画結果. $rgba = (0.5, 0.5, 1.0, 1.0)$ を基準色とし, 算出したエネルギー量を乗算して色彩を決定している.

更可能である. 図 4.11 は螺旋形状に補正を加えた際の描画結果である.

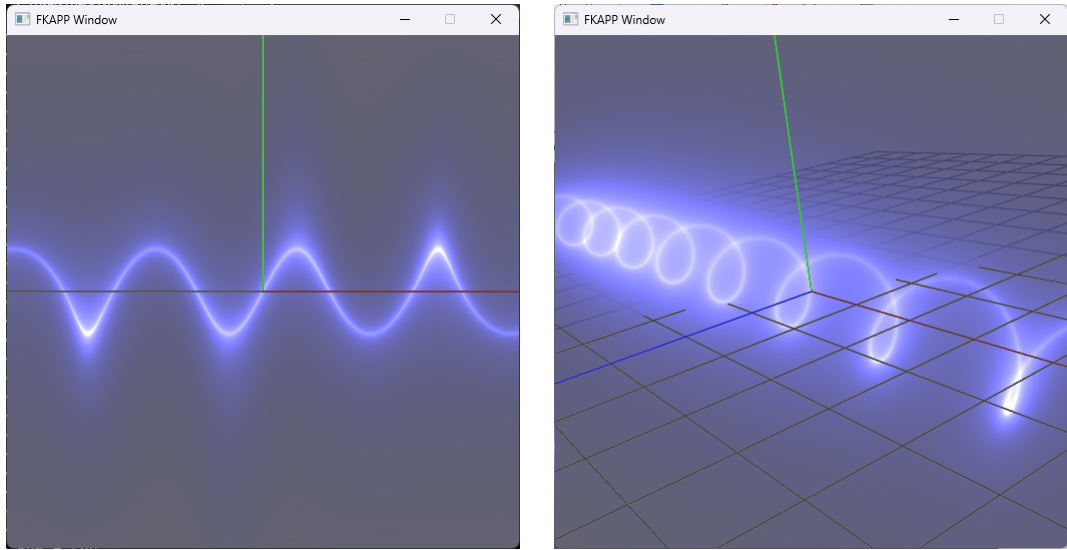


図 4.3: カメラ位置を変更した出力結果比較.

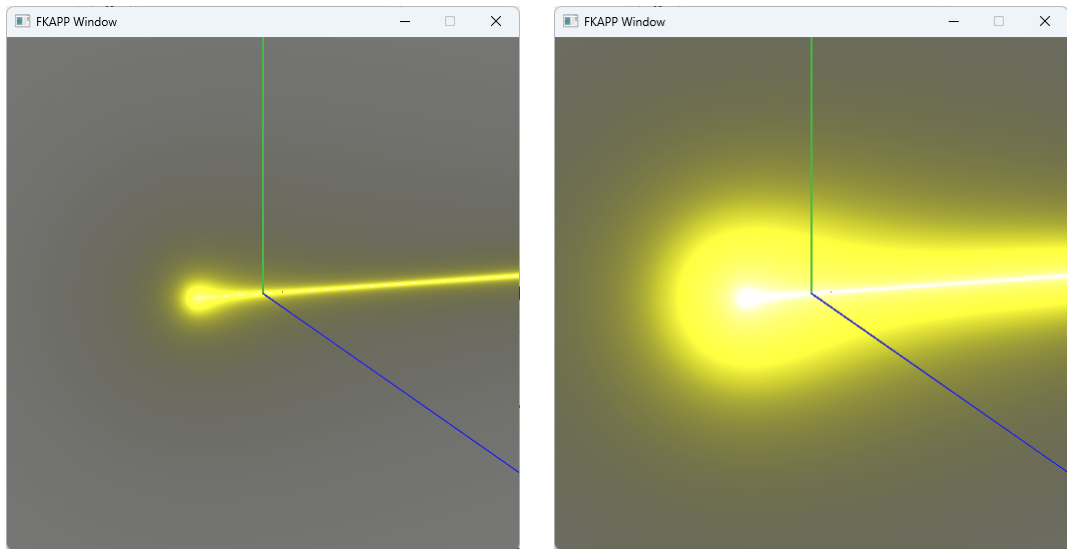


図 4.4: 調整係数 α を変更した出力結果比較.

4.3.4 移動表現による描画結果

移動表現に関する描画結果を示す。解析手法においては、任意に設定した球体の中心位置を変更する事でエネルギー波が目標地点へ向いながら移動していく様子を表現する。図 4.12 は解析手法における球体とカプセルの合形状に対し、任意中心地点を変更する事で進行方向へエネルギー波が伸びていく様子を示す。数値手法においては、積分区間を任意の範囲を増減する事で、エ

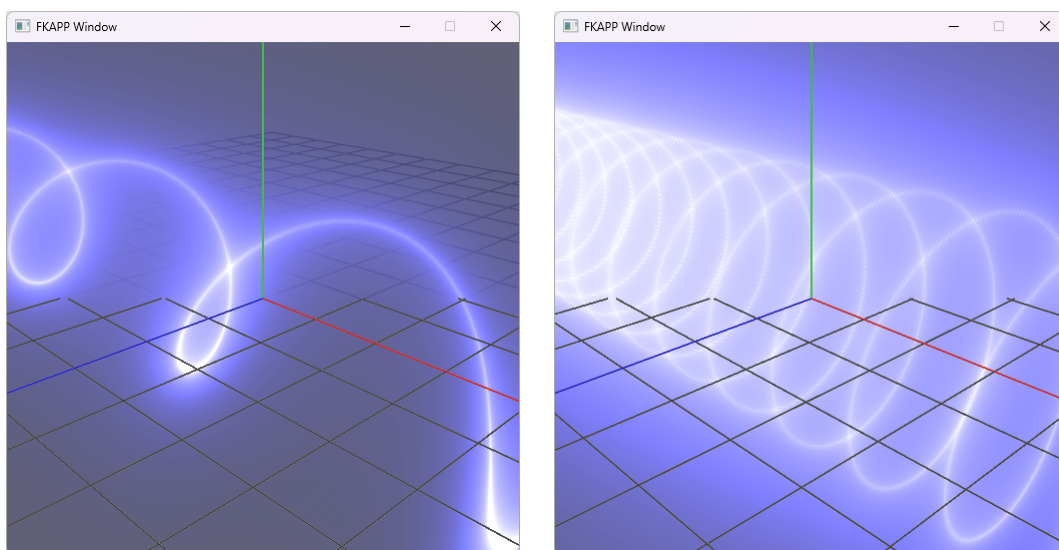


図 4.5: 調整係数 γ を変更した出力結果比較.

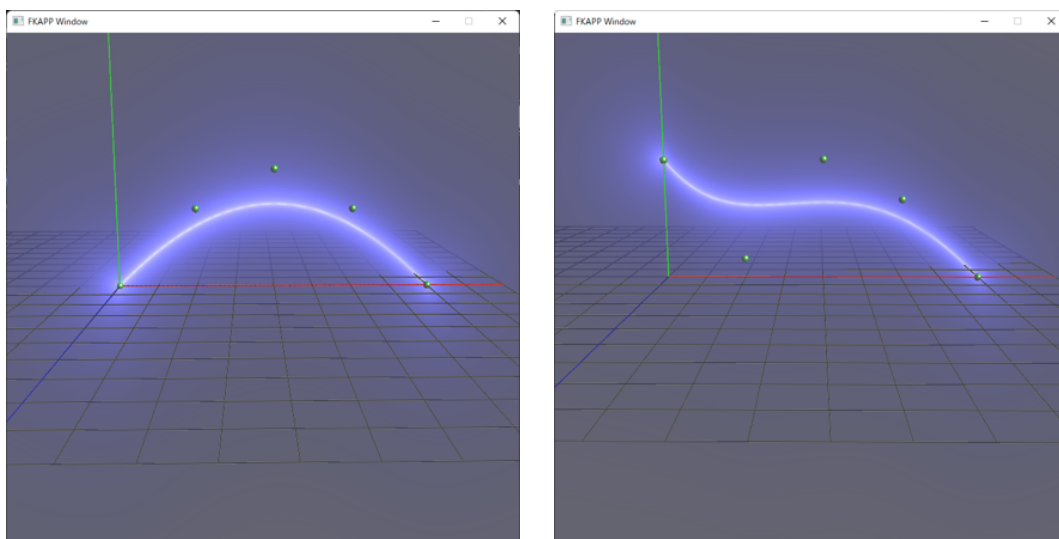


図 4.6: ベジエ曲線の制御点位置を変更した出力結果比較.

エネルギー波形状全体に対して描画する割合を変更可能となる．この描画する割合を時間変化に応じて制御する事で，エネルギー波の移動表現としている．図 4.13 は数値手法で生成したベジエ曲線に対し，時間変化に応じて積分区間を増加する事で進行方向へエネルギー波が伸びていく様子を示す．

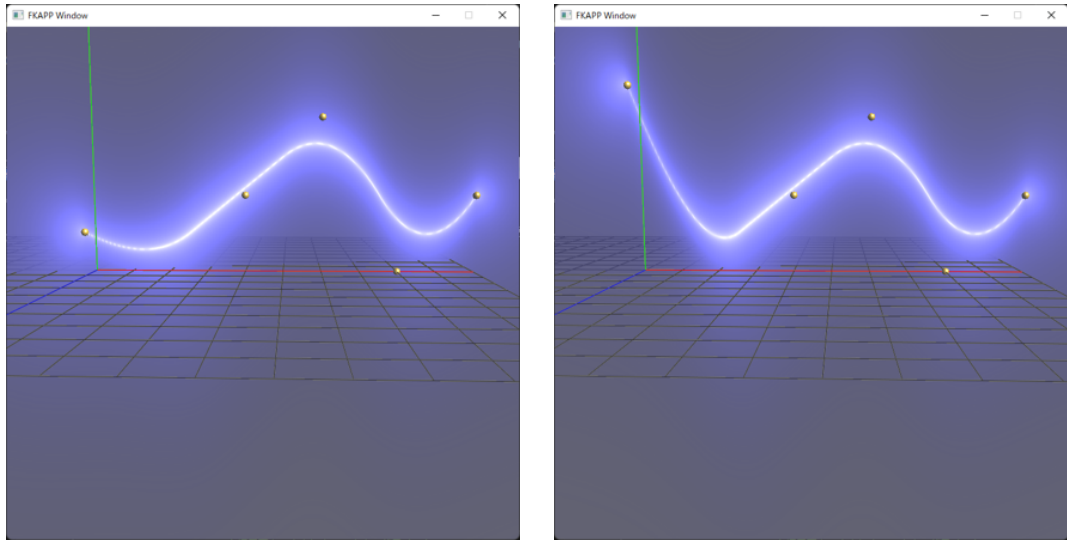
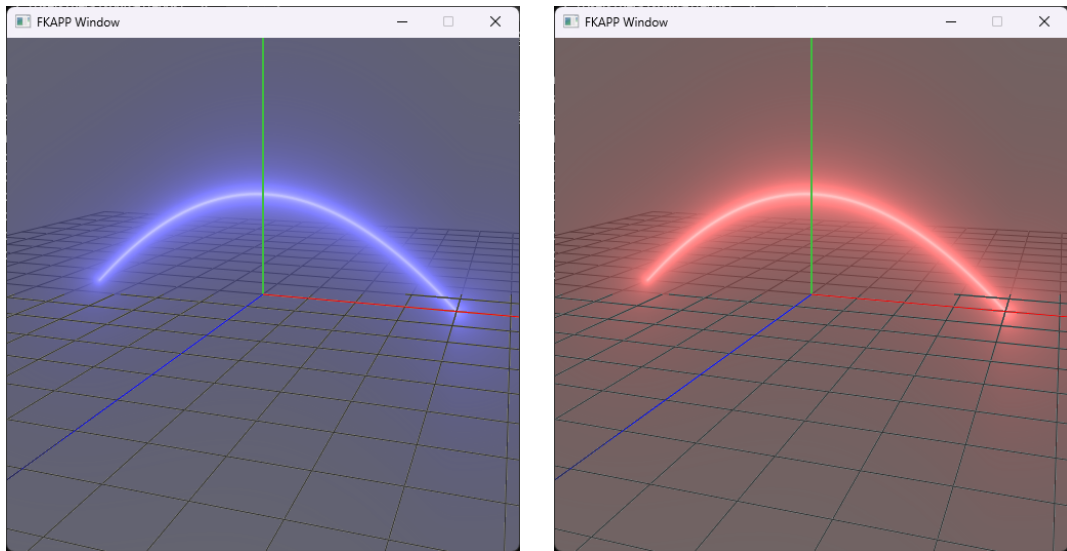


図 4.7: B-Spline 曲線の制御点位置を変更した出力結果比較.



(a) $(r, g, b, a) = (0.5, 0.5, 1.0, 1.0)$

(b) $(r, g, b, a) = (1.0, 0.5, 0.5, 1.0)$

図 4.8: 基準色に応じた色変化の様子.

4.3.5 隠面処理による描画結果

球体やカプセルといった解析手法で表現可能な形状における隠面処理について描画結果を示す。図 4.14 はエネルギー波と他オブジェクトをシーン中に配置し、エネルギー波の中心座標値が変動しているシーン描画である。エネルギー波のパラメータを $\alpha_s = \alpha_c = 20$ として、任意点 \mathbf{M} の z

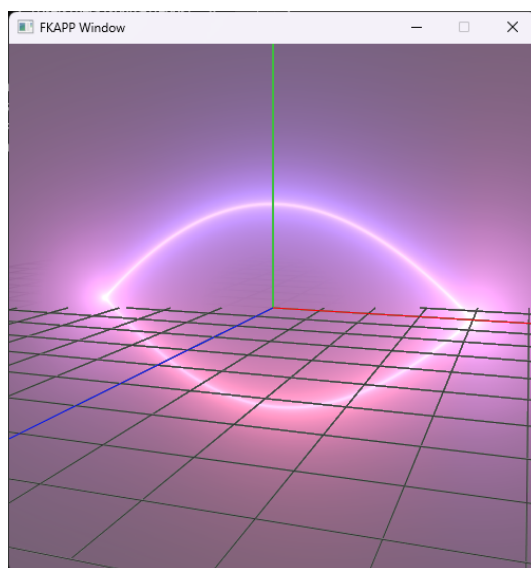


図 4.9: 異なる基準色曲線の計算結果合算の様子.

成分のパラメータ m_z を動的に変更した. エネルギー波以外のオブジェクトとして正 6 面体形状を 2 つ配置し, 一方を $(0.0, 0.0, -20.0)$, もう一方を $(10.0, 0.0, 10.0)$ に配置した. 描画面面奥側から手前側へかけて移動している様子となる. エネルギー波の投影面はカメラの直ぐ前方に配置することで, 隠面処理のみの効果によりエネルギー波と他物体との前後関係描画を実現している事を示す.

4.4 実行速度

プログラムの実行速度を 1 秒あたりの再描画回数である Frame Per Second(FPS) とし, 異なる解像度の投影面で計測を行った. 計測はプログラム実行後一定時間経過後の値である. 隠面処理には OpenGL の Frame Buffer Object(FBO) 機能を利用して実装を行った. デプス値を FBO に格納し, 投影面に対するフラグメントシェーダー内でデプス値を利用するマルチパスレンダリングを行っている. 表 4.3 は解析手法の実行速度を示す. 解像度の変化に対して処理速度の大幅な変化は確認出来ない. これは検証に用いた GPU 自体の性能が良く, FullHD($1920 * 1080$) を超える画素数の場合でも安定して高速な描画が可能であった. 隠面処理を施した場合, 単純にレンダリングパスが 1 つ増えている分だけ速度低下が確認できた. しかし処理有りの場合でも解像度の変化に対して処理速度の低下は確認できず, 解析的手法の安定性の高さを示す形となった.

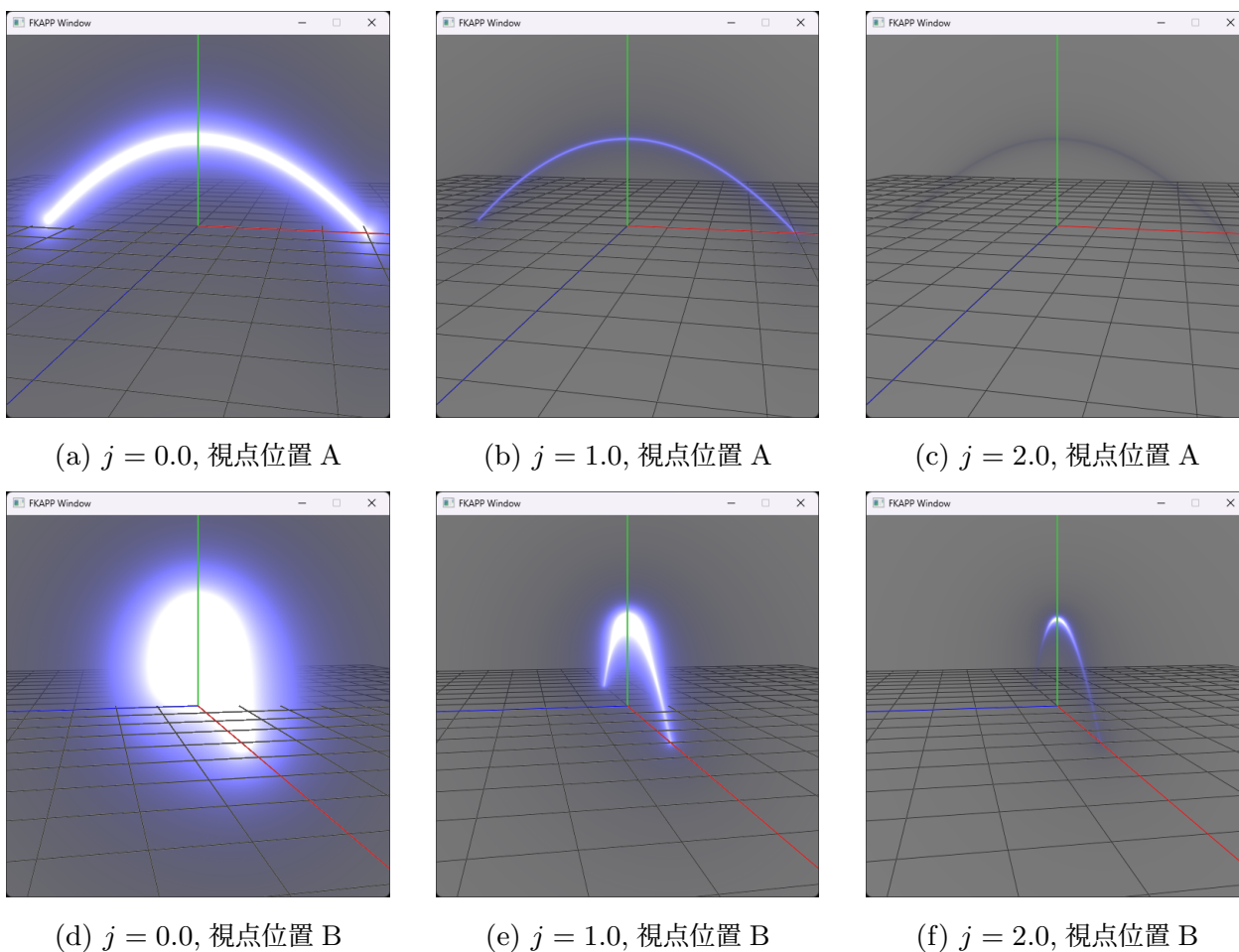


図 4.10: 減衰式を適用した曲線表示例. (a),(d) の $j = 0.0$ は減衰無しとなる. 減衰のパラメータ j 以外は (a) ~ (f) まで全て同値である.

表 4.4 は数値手法において、積分の分点数削減は行わず区間によらず一定の場合 ($n = 1$) の実行速度を示す. 三角関数ベースの曲線 2 つについて、高解像度かつ分点数増加に応じた処理速度低下が確認できたが、一般的なリアルタイムレンダリングが必要とする 60FPS 程度は概ね確保できる結果となった. パラメトリック曲線 2 つについては高解像度または分点数増加時には FPS の低下が顕著に確認できる. 三角関数ベースに比べ、パラメトリック曲線では曲線上の点を算出する計算工程や必要となるパラメータが多い分、処理速度低下量が多いと考えられる. B-Spline 曲線の実装上の注意として、ノットベクトルをグローバル変数化する必要がある. 本論文で用いたプログラミング言語である GLSL では再帰関数が利用不可なため、B-Spline 曲線の実装は多段の関数呼び出しによって実現した. そのため、シェーダーでの実行速度が他の曲線と比較して遅く

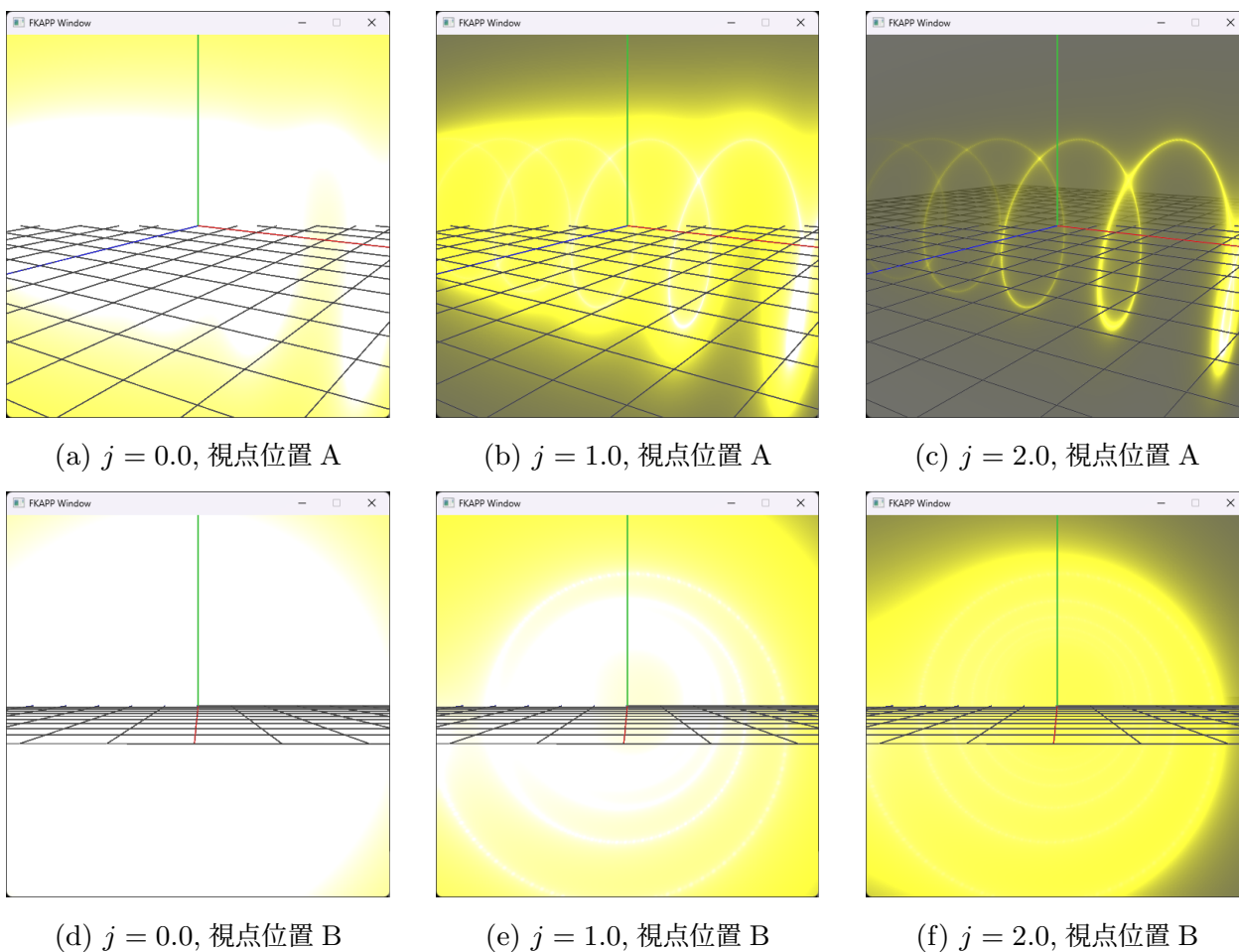


図 4.11: 減衰式を適用した螺旋表示例. (a),(d) の $j = 0.0$ は減衰無しとなる. 減衰のパラメータ j 以外は (a) ~ (f) まで全て同値である.

なったと考えられる.

視線とバウンディングボックスの位置関係に応じて分点数を制御した場合の実行速度を表 4.5 に示す. 表中の m と n はそれぞれ, 3.4.4 項で述べた分点数と分点削減率である. 円柱と 2 点で交差する視線数が投影面領域中の凡そ 5 割程度の場合と, 投影面全域に渡る場合とで計測を行った. 解像度は 2048 を用いている. 円柱との交点数が 1 以下の視線は自動的に分点削減対象となる. 三角関数ベース形状として螺旋曲線, 制御点によるパラメトリック形状として B-Spline 曲線で分点数削減を行った. 削減率に応じて処理速度が向上している事が確認できた. 2 点交差視線が投影面中に占める割合が少ない程処理全体で削減対象分点数が多くなり, その結果処理工程が少なくなる. 処理速度の検証結果からもその傾向が確認できた.

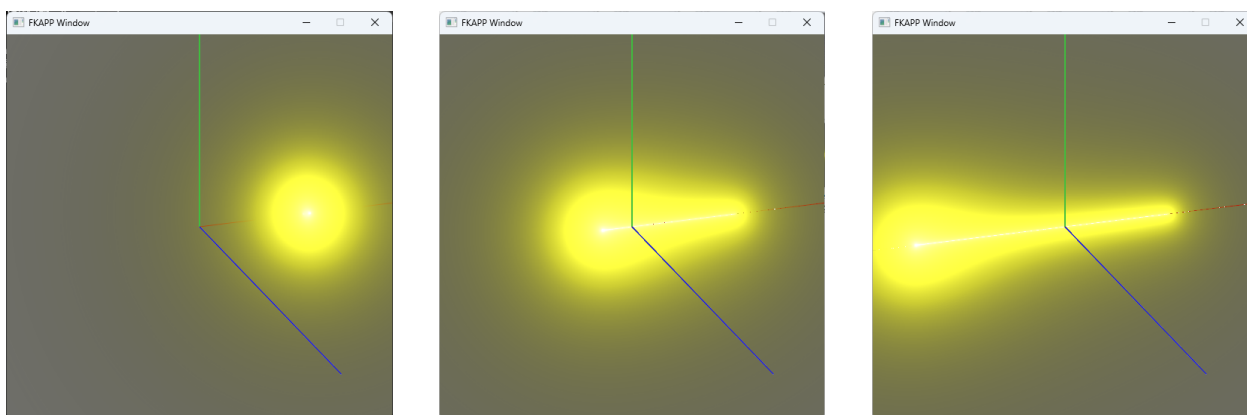
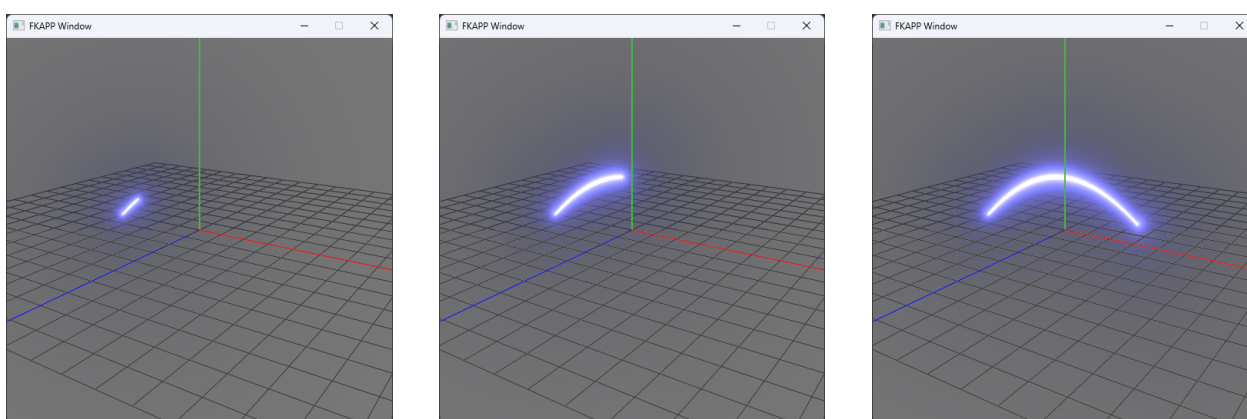


図 4.12: 時間変化に応じた中心位置変更によるエネルギー波の伸長表現.



(a) $u_t : 0 \sim 0.1$

(b) $u_t : 0 \sim 0.4$

(c) $u_t : 0 \sim 0.8$

図 4.13: 時間変化に応じた積分区間変更によるエネルギー波の伸長表現.

高速化処理を適用した際の表現上の特徴として、曲線形状を遠目から俯瞰的に見るカメラ配置の場合、バウンディングボックスを境にエネルギーの分布状態の切れ目が確認できる。分点数削減を行ってもエネルギー波形状自体は確認が出来るが、エネルギー波形状周辺空間のぼやとした分布が離散的な表現となってしまっており、空間中の稠密なエネルギー分布としては望ましくない。図 4.15 はカメラ位置を遠方に置き、分点削減の有無による螺旋曲線の描画結果を比較した画像である。図 4.16 はカメラ位置を遠方に置き、分点削減の有無による B-Spline 曲線の描画結果を比較した画像である。

近くから見るカメラ配置の場合は、出力結果上の微細な差異はありつつもほぼ分点削減前と遜色ない出力を得ることが可能である。図 4.17 はカメラ位置を近辺に置き、分点削減の有無による

表 4.3: 解析手法 - 実行速度 (FPS)

球体			
隠面処理	512*512	1024*1024	2048*2048
無	811.6	827.3	835.5
有	642.0	661.3	677.9
カプセル			
隠面処理	512*512	1024*1024	2048*2048
無	802.5	788.1	848.6
有	659.8	628.6	677.5
球体 + カプセル			
隠面処理	512*512	1024*1024	2048*2048
無	823.7	831.4	855.2
有	652.3	648.7	673.1

描画結果を比較した画像である。

解析手法は解像度の変化によらず、常に安定して高速であった。数値手法はサンプリングポイントが少ない場合、一部解析手法よりも高速な状況下が確認できた。解析手法は原始関数が初等関数の組み合わせで構成できるが、式の構成は複雑である。対して数値手法の計算式では、単に対応付けを行った2点間の距離計算が主な構成内容となる。その為、それぞれの積分式の複雑さに起因して、結果的に1視線あたりの計算回数が多い数値手法の方が高速になったと考えられる。

4.5 表現例の比較

本手法で作成例と既存のコンテンツ事例とを比較する。

カプセル型と球体型の組み合わせで行う端点膨張型直線形状は、多くのエネルギー波表現に使用されている形状である。図 4.19 はアニメコンテンツ内での表現例と本手法での作例比較である。

螺旋曲線は主に、ドリルの様な螺旋形状を放出することで敵を穿つさまを示したり、竜巻の様な形状や模様で利用される。図 4.19 はアニメコンテンツ内で螺旋形状が象徴的な攻撃手法と本手法での作例比較である。

次に制御点を用いた曲線形状について示す。エネルギー波は攻撃対象目標へ向けて移動する際に曲線軌道を描く表現も多い。コンテンツ上でエネルギー波を多量に放出する技を散見する事が

表 4.4: 数値手法 - 実行速度 (FPS)

螺旋			
分点数 m	512*512	1024*1024	2048*2048
100	1161.1	990.2	254.3
300	1163.0	341.2	86.5
500	810.5	206.1	52.0
三角関数和			
分点数 m	512*512	1024*1024	2048*2048
100	1144.7	1126.0	310.1
300	1159.8	419.8	106.0
500	972.0	252.6	64.0
4次ベジエ			
分点数 m	512*512	1024*1024	2048*2048
100	1155.0	691.2	175.1
300	915.0	233.9	59.0
500	555.7	140.8	36.0
B-Spline			
分点数 m	512*512	1024*1024	2048*2048
100	469.0	110.2	30.8
300	158.9	35.8	10.4
500	96.0	21.0	6.4

出来るが、手数が多さを演出するにあたりエネルギー波同士が重複しない様な形状や軌跡を採る事が多い。本手法の適用例であるベジエ曲線や B-Spline 曲線はこうした曲線形状、曲線軌道に適している。図 4.20 はアニメコンテンツ内で多量のエネルギー波で相手を攻撃する様子と本手法での作例比較である。カメラ位置やエネルギー波の量や形状といった要素は厳密には違いはあれど、本手法が手数の多い攻撃演出にも対応している様子が確認できる。

4.6 課題

実装内容より現行の課題点について述べる。解析的積分手法、数値的積分手法ともリアルタイムな描画を実現し、各種変更可能なパラメータに応じた表現変化を確認した。

解析的手法について、安定したリアルタイム描画を実現したが、表現可能な形状種が単純形状に限られている。また、分布の基点となる地点からのエネルギー値を視線上で正確に算出

表 4.5: 分点数削減を伴う実行速度 (FPS)

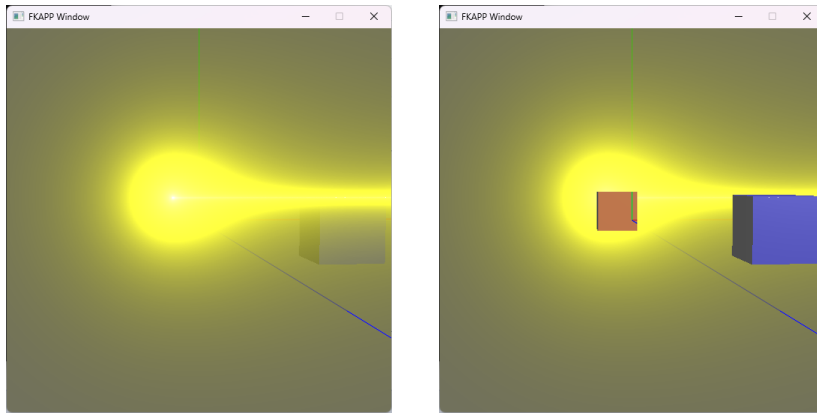
螺旋				
m	n	2点交差視線	FPS(削減前→後)	
100	2	5割程度	254.3	→ 438.5
	4			→ 757.0
	2	全域		→ 319.0
	4			→ 386.0
500	2	5割程度	52.0	→ 92.5
	4			→ 161.0
	2	全域		→ 66.0
	4			→ 80.0
B-Spline				
m	n	2点交差視線	FPS(削減前→後)	
100	2	5割程度	30.8	→ 65.0
	4			→ 114.0
	2	全域		→ 42.0
	4			→ 51.0
500	2	5割程度	6.4	→ 13.0
	4			→ 23.0
	2	全域		→ 9.0
	4			→ 11.0

可能な反面、視点から遠い地点にある感数値に対して減衰効果が適応不可となってしまう。目標とするエネルギー波形状の輪郭領域が、視点位置によっては想定以上に肥大化してしまう場合がある。

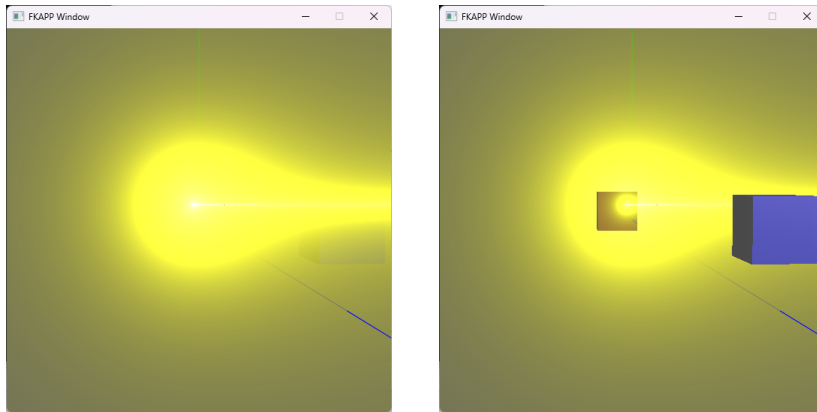
数値的手法について、リアルタイムな処理速度で一般陽関数曲線を表現する事に成功はしたが、各種調整係数や分点数 m によっては数値積分特有の離散的な出力状態が確認出来る。球や円柱形状で成功していた様な、解析的あるいはそれに近い精度の積分方法の更なる追求が必要である。離散的箇所を少なくしつつ高速な処理速度の実現の為、エネルギー波形状に寄与が低いと思われた領域の分点数を削減した方針も適用した。分点数削減によって処理速度の高速化が可能となった反面、カメラや対象曲線の位置関係、削減率や投影面解像度といった複数要素からの影響で離散的な描画結果が出現しやすくなった。その際、バウンディングボックスは円柱形とした。これはエネルギー波の形状や移動方向がある程度の単一方向へ伸びていくという特性を加味しつつ、線

分との交点計算速度を優先する為の方針であった。曲線の一部だけが極端にはみ出した様な形状の場合、包含しなければならない領域が多くなり、エネルギー波形状付近以外の領域に冗長な隙間が生じる結果となった。形状に沿った複雑なバウンディングボックスを設定することも可能であるが、物体の干渉判定にバウンディングボックスを用いる際の問題と同様に計算速度や精度とのトレードオフになる問題を孕んでいる。また、バウンディングボックスを境に分点数削減の影響がはっきりと描画されていた。境目で一律に切り替える方針よりも、境目付近で徐々に削減率を変動させる方針が改善案として考えられる。

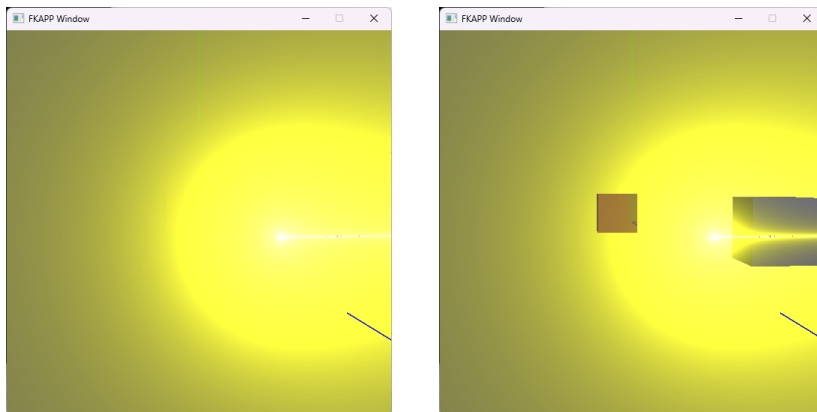
両手法は視点位置に応じてエネルギー値の積分結果が異なる方針となっている。その為、エネルギー波と視点位置が近距離であったり、エネルギー波の単調増加方向と視線方向が並行に近い位置関係の場合、各視線上のエネルギー値は高くなる傾向にある。その結果エネルギー波形状が想定よりも肥大化した表示結果となるケースが確認できた。数値的手法ではエネルギー値減衰もパラメータによって調整可能である為、視点位置に応じて本パラメータを動的に変更することで特定視点位置からの肥大化を低減することが対応策として考えられる。また、両手法とも大域的な分布状態を描画する事には適しているが、部分的な要素の描画には不向きである。エネルギー波は本体の周りに装飾的なパーツを纏わせる事が多いが、装飾的パーツは離散的であったり、ハンドメイドな意匠性が高く関数化には不向きである。



(a) エネルギー波位置の z 成分 = -35.0.

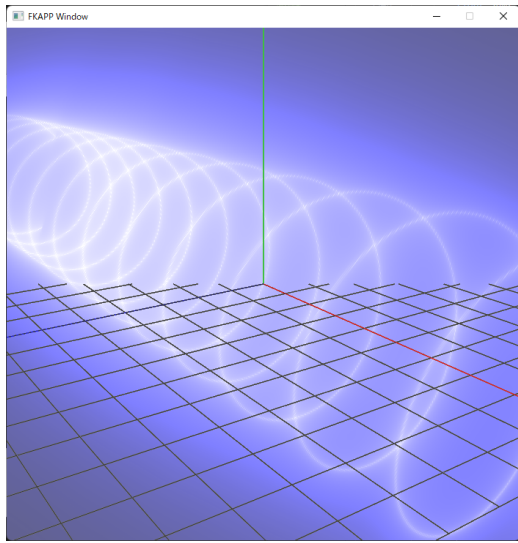


(b) エネルギー波位置の z 成分 = 5.0.

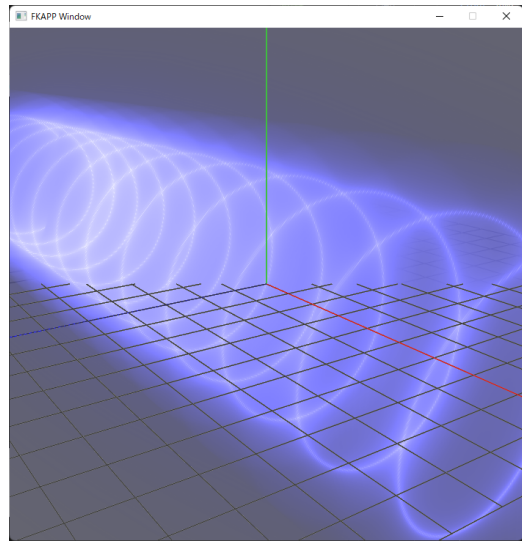


(c) エネルギー波位置の z 成分 = 35.0.

図 4.14: 隠面処理の有無による表示比較. 左が隠面処理無し, 右が隠面処理有りとなる.

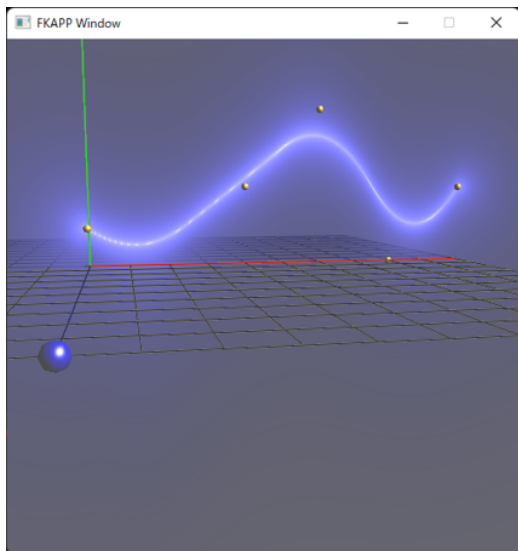


(a) 分点削減：未適用

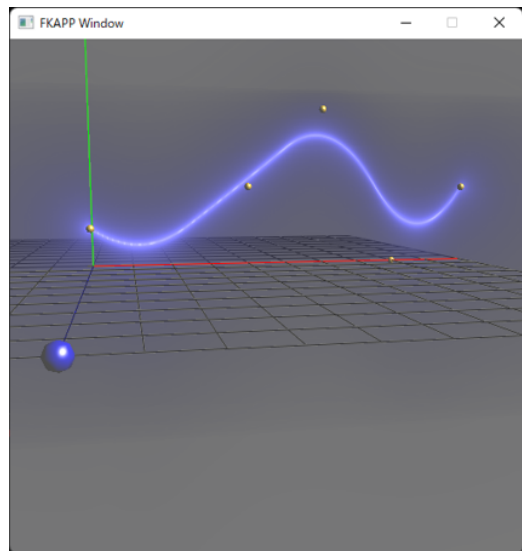


(b) 分点削減：適用後

図 4.15: 遠方カメラ配置 - 分点削減有無による螺旋曲線比較

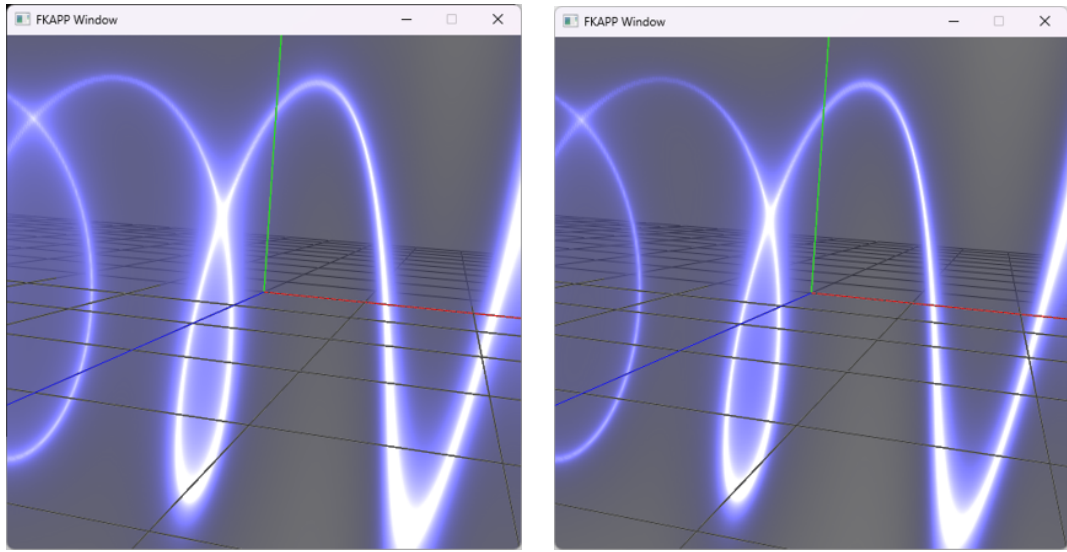


(a) 分点削減：未適用



(b) 分点削減：適用後

図 4.16: 遠方カメラ配置 - 分点削減有無による B-Spline 曲線比較



(a) 分点削減：未適用

(b) 分点削減：適用後

図 4.17: 近辺カメラ位置 - 分点削減有無による結果比較

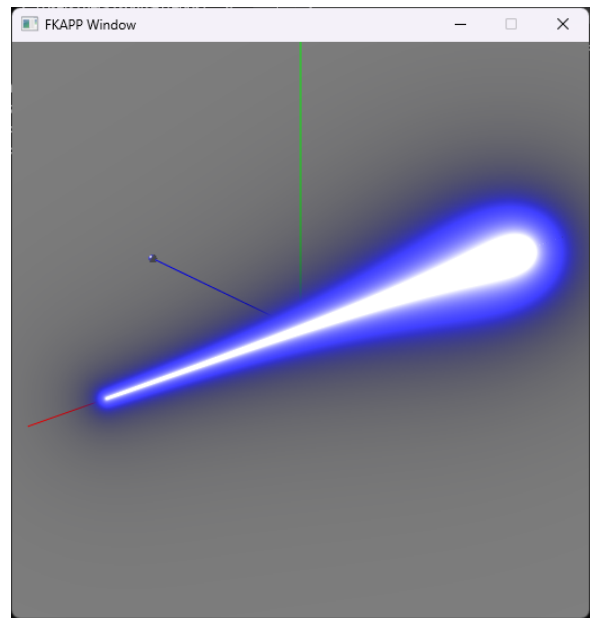


図 4.18: アニメコンテンツおよび本手法との作例 - 端点膨張型直線.

出典 “ドラゴンボール超” © 原案 鳥山明 / 集英社・フジテレビ・読売広告社・東映アニメーション

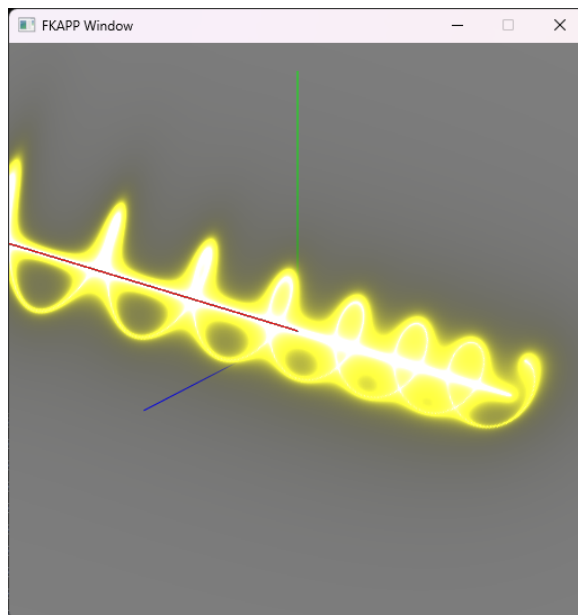


図 4.19: アニメコンテンツおよび本手法との作例 - 螺旋曲線. 中心の直線はベジェ曲線を用いて追加.
出典 “ドラゴンボール” © バードスタジオ/集英社・フジテレビ・東映アニメーション

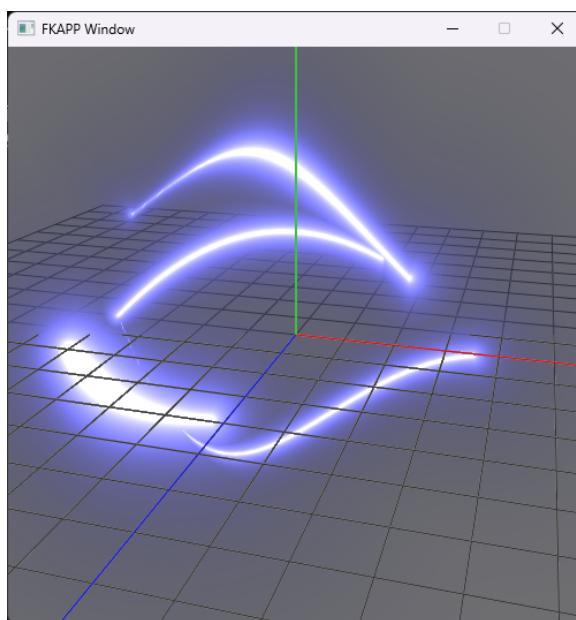


図 4.20: アニメコンテンツおよび本手法との作例 - 制御点による曲線.
出典 “葬送のフリーレン” © 葬送のフリーレン製作委員会 / 小学館・日本テレビ・TOHO animation

第 5 章

結論

5.1 研究のまとめと成果

本研究はエネルギー波という特徴的な視覚効果に着目し、関数分布を用いたエネルギー波エフェクトのリアルタイムレンダリング手法の構築を行った。

1章ではゲームにおけるエフェクトの役割や種類を説明すると共に、エネルギー波表現上の要素についてまとめた。近年のゲームエフェクト制作における主要表現技術を提示し、その課題点も併せて指摘した。

2章ではエネルギー波表現技術に関する調査について述べた。エネルギー波表現においては空間中の稠密な分布状態を描画する事が望ましいが、既存技術の多くは物体のサーフェスに焦点を当てている為、空間中の稠密な状態をリアルタイム描画する事に適していない。内部構造描画に焦点を当てているボリュームデータもインタラクティブなデータ変化には不向きな為、ゲームコンテンツ利用を想定するとやはりエネルギー波表現に用いる事は適していない。1章で述べた既存のエフェクト作成ツールでの作成方針特徴と合わせて、本研究が解決すべき課題を提示した。

3章では提案手法について述べた。目標形状に対して提案手法は2種類に分類した。3.2節では理論の全体像について解説し、続く3.3節では単純形状に適した解析的積分手法、3.4節では曲線形状に適した数値的積分手法についてそれぞれ提案を行った。

解析的手法では球体や円柱、それらを組み合わせた形状の描画理論を提示した。任意の基準地点をエネルギー分布の中心地点と定め、その基点からの距離に応じた分布状態を基底した。基底した関数を解析的に積分可能な原始関数を求め、安定した高速描画を実現した。積分区間の制御を用いる事で、球体と円柱を組み合わせたカプセル形状の描画、デプス値を利用した隠面処理といった表現も可能となった。

数値的手法では、エネルギー波が持つ目標地点に向かう進行軸方向へは単調増加する特性を利用し、一般陽関数を定義した。この条件を満たす曲線式を一般陽関数曲線と定め、エネルギー波分布関数として採用した。視線上のある地点と一般陽関数曲線上のある地点とを媒介変数で一對の組として結びつけ、結びつけた媒介変数に対する積分式を提案した。数値的手法では表現可能な曲線形状として螺旋形状や3次以上の高次ベジェ曲線形状、B-Spline曲線形状といった汎用性の高い曲線形状の描画が可能となった。また、数値的手法では積分区間の制御によってエネルギー

波の移動表現を実現した。

4章では、ここまで述べた手法を実際に実行し、描画結果や処理速度についての検証を行い、本研究の有用性について述べた。

本研究によって得られた成果をまとめた結果、以下の結論が得られた。

1. 空間中に稠密なエネルギー波表現を、リアルタイムグラフィックス技術によって視点の位置や姿勢の制約を持たない状態で描画することを可能とした。
2. 既存コンテンツを参考に本研究での目標形状を定め、各形状に応じたエネルギー分布関数を提示、2つの異なる積分方針を示した。
3. 球体や円柱といった単純形状では解析的積分手法により、高解像度な投影面であっても処理速度を一定に保つ事が可能な、高速性と安定性を兼ね備えた描画結果を得ることに成功した。
4. 一般陽関数曲線といった複雑な形状では数値的積分手法により、これまでリアルタイム描画が不可能であった高次 Bézier 曲線や B-spline 曲線、螺旋曲線といった形状を実現した。
5. 各積分手法は OpenGL のシェーダー機能で並列計算が可能な形式である為、手法全体として各種パラメータ調整による分布状態のインタラクティブな変更を踏まえた上で実用的に高速な描画速度を実現した。

以上の結果、本研究で目的としていたエネルギー波表現のリアルタイムレンダリングに関する問題を解決し、これまで不可能であったエネルギー波形状種類に対して動的な視点変更や形状変形を実現した。

5.2 現状の課題

本研究により、空間中のエネルギー分布関数を用いてレンダリングする、エネルギー波表現の新たなレンダリング方針を確立した。単純形状では高速に安定した描画手法を、複雑な曲線形状ではこれまで不可能であった形状種をリアルタイムに表現可能となった。

現状の手法において大域的な形状変形や移動表現に向いているが、飛沫や凹凸といった局所的

な装飾的表現は依然として困難である。コンテンツにおけるエネルギー波表現において、放出しているエネルギー本体は稠密かつ連続的な分布による見た目をしているが、本体周辺の装飾的表現は離散的である事がほとんどである。こうした装飾的表現も関数化方針に組み込む事が出来れば、更に意匠性の高いエネルギー波表現をデザイン可能となる。稠密な分布状態を表現可能となったが、基準点からは一様な変化となっている。稠密性を保ちつつ、局所的には非一様な分布関数に拡張する事で、こちらも意匠性の向上が期待できる。エネルギー波を発光体として考えた場合、既存コンテンツ内ではエネルギー波の中心にライトを設定して他オブジェクトへのライティング効果も実現している。本手法ではエネルギーの強さによって形状自体が変化する為、形状変化に応じてライティング効果も調整する必要があると考える。エネルギー波は他オブジェクトや他エネルギー波と衝突すると、大きく形状が変化する描写も多い。こうした相互干渉による形状変化についても、今後の課題である。

5.3 今後の展開

本研究はアニメや特撮で用いられていた特殊効果を、数式を用いてゲームエフェクトへ落としこむ方針と捉える事ができる。この方針には更なる発展性が期待できる。まず、エネルギー波以外にも類似の視覚表現がある。「オーラ」と呼ばれるエフェクトであり、キャラクターの体の周囲に光の模様を纏わせる事で、特別な力強さを視覚化する表現手法である。エネルギー波と同様に光の強さによってその強弱を表現しており、大抵の場合エネルギー波とセットでキャラクターに付与する。こうしたオーラエフェクトに対しても、数式を用いた基底やアニメーション変化の実現が期待できる [49]。アナログな工程でアニメ制作が行われていた際、エネルギー波の撮影には専用の光源を設置していた。撮影の際に媒質と呼ばれる素材を重ねる事で、光に色をつけたり透過・屈折によって生じる光学模様をエネルギー波表現に用いていた。こうしたアナログ時代の撮影工程を 3D 空間中でシミュレート [50] する事で、アナログ時代のケレン味のあるエネルギー波表現をゲームコンテンツ内でリアルタイムに表示可能になる事が期待できる。

謝辭

本研究を完遂するにあたり長期間に渡り多くの方々のご指導・ご鞭撻を頂戴しました。

学部・修士含めて研究の礎をご指導ご教授頂きました，渡辺大地先生に感謝いたします。研究指針や論文執筆，発表等，研究活動全てにおいて常に指針を提示して頂きました。研究以外での時間も多く共有させて頂き，「先生」という言葉は当方の中では渡辺先生を指す言葉となっております。

本研究の主査を引き受けて下さいました三上浩司先生に感謝いたします。三上先生には学士時代からの指導補助や，修士時代にも副査を務めて頂きました。現場感覚に近い三上先生の知見は，いつも研究発想の種となり，研究結果へのモチベーション維持の柱となっております。今回も引き続きご指導頂けて，万感の思いと共に改めて御礼申し上げます。

審査にあたりご多用の所副査を引き受けて下さいました近藤邦雄先生，菊池司先生，竹島由里子先生に感謝いたします。近藤先生におかれましては三上先生と同様に修士時代にも副査を務めて頂きました。今回，近藤先生の退職後にも関わらず副査を依頼させて頂いたのは，私たつての希望でございました。いつも暖かく時に厳しく背中を押し続けて頂けた事が，本研究活動の原動力となっております。菊池先生におかれましては，VFX 研究の先達としてさまざまご意見頂戴いたしました事感謝申し上げます。また，当方在職中の折には業務以外でのご配慮いただきました事も加えて御礼申し上げます。竹島先生におかれましては，審査の折にきめ細やかな査読，ご意見頂戴いたしました事感謝申し上げます。

メディア学部の諸先生方に感謝申し上げます。学部生時代に先生方から学んだメディア学が血肉となり，今回の成果を上げるに至りました。

annkw，三四郎 ANN0，マヂラブ ANN0 に感謝します。1人作業が続く中，いつも私の心に活力を与えてくれました。

最後に，いつも私を支えてくれた両親と家族，友人たちに感謝いたします。

参考文献

- [1] Craig Donner and Henrik Wann Jensen. Light diffusion in multi-layered translucent materials. *ACM Transactions on Graphics*, Vol. 24, No. 3, pp. 1032 – 1039, 2005.
- [2] Gaurav Bhokare, Eisen Montalvo, Elie Diaz, and Cem Yuksel. Real-Time Hair Rendering with Hair Meshes. *ACM SIGGRAPH 2024 Conference Papers*, pp. 1 – 10, 2024.
- [3] FOUNDRY. Nuke. <https://www.foundry.com>. 参照: 2024-7-23.
- [4] Yongning Zhu and Robert Bridson. Animating sand as a fluid. *ACM Transactions on Graphics*, Vol. 24, No. 3, pp. 965 – 972, 2005.
- [5] 越塚誠一. 粒子法シミュレーション 物理ベース CG 入門. 培風館, 2008.
- [6] 大塚航輝. CC2 流! ビジュアルエフェクトアーティスト入門講座. https://cedil.cesa.or.jp/cedil_sessions/view/2858. 参照: 2024-7-23.
- [7] 任天堂. エフェクトデザインとは? <https://www.nintendo.co.jp/jobs/introduction/design/work04.html>. 参照: 2024-7-23.
- [8] 野村宏平. ゴジラ大辞典. 笠倉出版社, 2004.
- [9] 円谷プロダクション. ウルトラマン大図鑑. ポプラ社, 2011.
- [10] 鳥山明. DRAGON BALL 大全集 - 鳥山全ワールド (7). 集英社, 1996.
- [11] Bandai Namco Entertainment. 「ドラゴンボール Sparking!ZERO」 - キャラクタートレーラー「永遠のライバル」 (0:23 - 0:25)(1:12 - 1:14). <https://www.youtube.com/watch?v=mh4Tw6xiGkA>. 参照: 2024-5-21.
- [12] TOHOanimation. フェルン vs エーレ/ 『葬送のフリーレン』第 20 話「必要な殺しよ」より (0:02 - 0:16). <https://www.youtube.com/shorts/CS3W4huVevM>. 参照: 2024-5-21.
- [13] KYORAKU CHANNEL. コンバトラー V のテーマ 2021ver (2:17 - 2:22). <https://www.youtube.com/watch?v=2Kh7uBEJbnE>. 参照: 2024-5-21.
- [14] ガンダムチャンネル. ノーベルガンダム — 昼 MS【ガンチャン】 (0:14 - 0:18). <https://www.youtube.com/watch?v=Qcjt3tyuN1U>. 参照: 2024-5-21.
- [15] Unity. Visual effect graph. <https://unity.com/ja/visual-effect-graph>. 参照: 2022-3-16.

- [16] Unreal Engine. Niagara ビジュアルエフェクトシステム. <https://docs.unrealengine.com/4.27/ja/RenderingAndGraphics/Niagara/>. 参照: 2022-3-16.
- [17] 桜井政博のゲームを作るには. ビルボード【エフェクト】. <https://www.youtube.com/watch?v=YzSh4H8EKRI>. 参照: 2024-7-23.
- [18] Effekseer 開発チーム. Effekseer. <http://effekseer.github.io/jp/index.html>. 参照: 2024.7.23.
- [19] Robert A. Drebin, Loren Carpenter, and Pat Hanrahan. Volume rendering. *SIGGRAPH'88, Computer Graphics*, Vol. 22, No. 4, pp. 65–74, 1988.
- [20] M. Groher, F. Bender, R. T. Hoffmann, and N. Navab. Segmentation-driven 2d-3d registration for abdominal catheter interventions. *Proceedings of the 10th international conference on MICCAI*, 2007.
- [21] K. Zhou, Z. Ren, S. Lin, H. Bao, B. Guo, and H. Y. Shum. Real-time smoke rendering using compensated ray marching. *ACM Transactions on Graphics*, Vol. 27, No. 3(Aug), pp. 36:1–12, 2008.
- [22] H. Tuy and L. Tuy. Direct 2d display of 3d objects. *IEEE mag. Computer Graphics and Applications*, 1984.
- [23] J. kruger and R. Westermann. Acceleration techniques for gpu-based volume rendering. *Proceedings of the 14th IEEE Visualization*, 2003.
- [24] 高棹大樹, 金井崇, 山口泰. GPU を用いた高品質ボリュームレンダリングに関する研究. 情報処理学会研究報告, Vol. 2007, pp. 67–72, 2007.
- [25] Mehmet Oguz Derin, Takahiro Harada, Yusuke Takeda, and Yasuhiro Iba. Sparse Volume Rendering using Hardsare Ray Tracing and Block Walking. *SIGGRAPH Asia 2021 Technical Communications*, No. 19, pp. 1 – 4, 2021.
- [26] JangaFX Software. Embergen. <https://jangafx.com/software/embergen>. 参照: 2024.7.24.
- [27] Z.PAN and D.MANOCHA. Efficient solver for spacetime control of smoke. *ACM Trans-*

- actions on Graphics*, Vol. 36, No. 5, pp. 162:1 – 162:13, 2017.
- [28] A.Stomakhin and A.Selle. Fluxed animated boundary method. *ACM Transactions on Graphics*, Vol. 36, No. 4, pp. 68:1 – 68:8, 2017.
- [29] D.Nowrouzezahrai, J.Johnson, A.Selle, D.Lacewell, M.Kaschalk, and W.Jarosz. A programmable system for artistic volumetric lighting. *ACM Transactions on Graphics*, Vol. 30, No. 4, pp. 29:1–29:8, 2011.
- [30] 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. 芸術科学会論文誌, Vol. 13, No. 3, pp. 144–153, 2014.
- [31] J.Blinn. A generalization of algebraic surface drawing. *ACM Transactions of Computer Graphics*, Vol. 1, No. 3, pp. 235 – 256, 1982.
- [32] H.Nishimura, M.Hirai, T.Kawai, I.Shirakawa, and K.Omura. Object modeling by distribution function and a method of image generation. *Journal of papers given by at the Electronics Communication Conference*, Vol. 568, pp. 718 – 725, 1985.
- [33] Y. Kanamori, Z. Szego, and T. Nishita. Gpu-based fast ray casting for a large number of metaballs. *In Proc. Eurographics*, 2008.
- [34] Y. Ohtake, A. G. Belyaev, M. Alexa, G. Turk, and H. P. Seidel. Multi-level partition of unity implicits. *ACM Transactions on Graphics (Proc. SIGGRAPH 2003)*, Vol. 22, pp. 463–470, 2003.
- [35] T. Kanai, Y. Ohtake, H. Kawai, and Kiwamu Kase. Gpu-based rendering of sparse low-degree implicit surface. *In Proceedings of GRAPHITE 2006*, pp. 165–171, 2006.
- [36] Qi Liu and Alexei Sourin. Function-based shape modeling and visualization in x3d. *Proceedings of the eleventh international conference on 3D web technology*, pp. 131–141, 2006.
- [37] A.Pasko, V.Adzhiev, A.Sourin, and V. Savchenko. Function representation in geometric modelling: concept, implementation and applications. *The Visual Computer*, Vol. 11, No. 8, pp. 429–446, 1995.

- [38] A. Pasko, V. Adzhiev, B. Schmitt, and C. Schlick. Constructive hypervolume modeling. *Graphical Models*, Vol. 63, No. 6, pp. 413–442, 2001.
- [39] B. Schmitt, A. Pasko, V. Adzhiev, and C. Schlick. Constructive texturing based on hypervolume modeling. *Journal of Visualization and Computer Animation*, Vol. 12, No. 5, pp. 297–310, 2002.
- [40] B. Schmitt, A. Pasko, and C. Schlick. *Constructive hypervolume modeling using extended space mappings*. Heterogeneous objects modelling and applications: collection of papers on foundations and practice. Springer-Verlag, Berlin, 2008.
- [41] T. Watanabe, M. Abe, and K. Konno. Real-time rendering technique for visual expression of arbitrary-shaped energy wave. *The Journal of the Society for Art and Science*, Vol. 15, No. 2, pp. 98 – 110, 2016.
- [42] Scott D. Roth. Ray casting for modeling solids. *Computer Graphics and Image Processing*, Vol. 18, No. 2, pp. 109 – 144, 1982.
- [43] 平山弘. Taylor 展開を利用した数値積分法. ハイパフォーマンスコンピューティング 研究報告, Vol. 2012, No. 23, pp. 1 – 6, 2012.
- [44] Marc Levoy. Efficient ray tracing of volume data. *ACM Transaction on Graphics*, Vol. 9, No. 3, pp. 245–261, 1990.
- [45] 松井学, 伊野文彦, 萩原兼一. 大規模データセットを可視化するための効率の良い並列ボリュームレンダリング. 情報処理学会論文誌, Vol. 45, No. SIG 11, pp. 346–355, 2004.
- [46] 小野讓二. 大規模数値計算における可視化技術の最先端. 可視化情報, Vol. 32, No. 125, pp. 52–57, 2012.
- [47] 日比野勤, 長谷川武光, 二宮市三, 細田陽介, 佐藤義雄. 二宮法と FLR 法の結合による新しい適応型積分. 情報処理学会論文誌, Vol. 44, No. 10, pp. 2419 – 2427, 2003.
- [48] G.E.Farin. NURBS 射影幾何学から実務まで 第 2 版. 共立出版, 2001.
- [49] M.Abe and T.Watanabe. Real-time rendering technique for visual expression of aura effect. *International Conference of Image Electronics and Visual Computing*, 2019.

- [50] M.Abe and T.Watanabe. Real-time rendering of energy wave representation using transmitted light. *International Conference of Image Electronics and Visual Computing*, 2024.

発表業績

学術論文

1. 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, 芸術科学会論文誌第 9 卷 (第 3 号), 2010.
2. 阿部雅樹, 渡辺大地, 一般陽関数曲線によるエネルギー波形状のリアルタイムレンダリング, 芸術科学会論文誌第 23 卷 (第 3 号), 2024.

国際会議口頭発表 (査読付)

1. Masaki Abe, Taichi Watanabe, Real-time rendering technique for visual expression of aura effect, IEVC2019, 2019.
2. Masaki Abe, Taichi Watanabe, Real-time rendering of energy wave representation using transmitted light, IEVC2024, 2024.

国内会議口頭発表 (査読付)

1. 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, NICOGRAPH2009, 2009.
2. 阿部雅樹, 渡辺大地, 螺旋曲線によるエネルギー波形状のリアルタイムレンダリング, NICOGRAPH2023, 2023.

国内会議口頭発表 (無査読)

1. 阿部雅樹, 渡辺大地, 関数分布を用いたエネルギー波形状の拡張に関する研究, 令和 3 年度 第 2 回芸術科学会東北支部研究会, 2022.
2. 阿部雅樹, 渡辺大地, テイラー展開を利用した関数分布によるエネルギー波形状の拡張に関する研究, 令和 4 年度 第 2 回芸術科学会東北支部研究会, 2023.