芸術科学会論文誌 Vol. 23, No. 3, pp. 8:1 – 8:14 (2024)

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

阿部雅樹 [1)](正会員)　渡辺大地 [2)](正会員)

1,2) 東京工科大学メディア学部

Real-time rendering of Energy-Wave using general explicit function curves

Masaki Abe[1)]　**Taichi Watanabe**[2)]

1,2) School of Media Science, Tokyo University of Technology

1)abemsk @ edu.teu.ac.jp，2)earth @ stf.teu.ac.jp

概要

漫画やアニメ，ゲームにはエネルギー波と呼ばれる表現が存在する．エネルギー波の描画表現では，強い発光を伴いながらエネルギーの塊が移動する様子が描かれている．筆者らはこれまで，プリミティブ形状においては解析積分可能なエネルギー分布を規定した．2 次曲線や円環形状を表現するエネルギー分布では，数値積分を用いて形状表現を行ってきた．本研究では，線積分における媒介変数と陽関数曲線における媒介変数の 2 つの対応を取る事で，1 つの関数として三角関数曲線やパラメトリック曲線といった分布状態を高速にレンダリングする表現する手法を提案する．螺旋曲線や B-Spline 曲線を対象に被積分関数を構築し，合成シンプソン法で積分する事でリアルタイムな描画速度を保ちつつ，実装検証により関数曲線を用いたエネルギー波形状の表示に成功した．

Abstract

Cartoons, animations, and video games contain representations known as energy waves. Energy waves represent a moving mass of energy accompanied by strong luminescence. The authors have defined analytically integrable energy distributions for primitive shapes and have used numerical integration for energy distributions representing quadratic curves and circular shapes. In this study, we propose a method to represent distributional states such as trigonometric curves and parametric curves as a single function that can be quickly rendered by taking two correspondences between the mediating variable in line integrals and the mediating variable in explicit curves. We constructed an integral function for spiral curves and B-spline curves and integrated it using the composite Simpson method to maintain real-time rendering speed.

1　はじめに

漫画やアニメ，ゲームには「エネルギー波」と呼ばれる表現が一般的に存在する．エネルギー波の描画表現では，強い発光を伴いながらエネルギーの塊が移動する様子が描かれている [1]．ビデオゲームにおけるエフェクトは，キャラクターの状態変化を可視化するといった視覚効果や，キャラクターによる特別な攻撃方法を示す言葉として広く用いられている．この攻撃方法の中には，エネルギーを光線状あるいは光弾状に放出する描写があり，俗にエネルギー波と呼ばれ愛好家の中では一般的な表現として親しまれている．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，本論文では漫画やアニメ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．図 1 は創作コンテンツ内におけるエネルギー波の一例である．

図 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，動きのアニメーションを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．一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[2]，UE4 では Niagara[3] が VFX 生成システムとして提供されている．これらのツールは特にパーティクルを用いたエフェクト表現作成に適している．パーティクルの集合でエフェクトを表現することで，炎や水といった自然現象に近いエフェクト表現が可能となる．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古くから炎や雷，水といった自然現象をモチーフにした表現が多く存在している．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，漫画的表現に向いた手法である．

本研究ではキャラクターの攻撃方法の中でも光線状に放出するエネルギー波エフェクトを対象にする．光線形状も直線や曲線など様々な表現が多くの作品中で見受けられる．本研究ではビデオゲームに用いられるエネルギー波表現に対して特に，曲線形状に着目した新たな関数分布表示方法を提案する．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，多くの創作コンテンツ内で用いられる．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」[4] や，「葬送のフリーレン」における「ゾルトラーク」[5]，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」[6] や，「機動武闘伝 G ガンダム」における「ビームリボン」[7] の様に，様々な作品において表現が行われている．

曲線形状を媒介変数による関数で表現し，曲線の媒介変数と線積分の媒介変数による 2 変数関数として空間中のエネルギー分布を規定する．更に媒介変数同士の対応を取る事で，形状特徴を保持したままの描画が可能となる．これまででは実現できなかった種類の曲線形状を実時間でレンダリングすることで，関数分布によるエネルギー波表現の拡張を目指す．

我々は本研究の初期成果を NICOGRAPH2023[8] で発表した．表現形状が螺旋に特化した限定的なものであるという指摘を受け，表現可能形状の拡充を行なった．提案手法が複数種類の曲線に対応可能である旨を体系化

した記述に変更し，検証を追加した．検証の結果，これまで実現不可能であった形状種類について十分な処理速度を保ちつつ描画可能となった．

2 関連研究

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 [9][10] 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 [11] 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．

不定形自然現象や一定領域内の稠密な光学現象にはボリュームレンダリングも良く用いられる．3 次元のボクセルデータを用いて物体の内部構造を含む可視化や不定形自然現象等，用途は多岐に渡り広く一般的に利用されている．ボリュームデータは空間中を一定間隔で区切り (一区画をボクセルと呼ぶ)，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．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画像が出力可能である．その反面，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，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．

2.1 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[12] や西村らによる Metaball[13] がその代表例であり，古くから多用されている．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粗密な濃度分布を描画する目的には直接的には不適当である．

2.2 関数を用いたエフェクト表現に関する研究

筆者らはこれまでも関数分布によるエネルギー波表現を行ってきた [14][15]．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．文献 [14] では球体やシリンダー形状等のプリミティブな形状表現を可能とした．描画面に対して視線を飛ばし，ピクセル毎に映るエネルギー量を解析的に求めることが可能である．文献 [15] ではトーラス型形状や 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，解析解を求めることが困難となり，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． プリミティブな形状は解析的積分が可能である反面，関数種類が非常に限定されエネルギー波形状創作上の表現力に乏しい．2 次ベジェ曲線や円環形状によって表現力は向上したが，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．既存手法の球や円柱，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，球や曲線の個数が増加すれば全体の処理速度が低下する．渡辺 [15] 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．

本研究ではこれまでリアルタイム表示が不可能であった分布形状のリアルタイムレンダリングを目標とし，三角関数曲線やパラメトリック曲線に着目した．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軌跡として用いられる事の多い表現種である．

3 提案手法

本章では提案手法について述べる．初めに基本的なレンダリングの方針について述べる．次に陽関数曲線を媒介変数表記し，分布関数用媒介変数と線積分用媒介変数同士の対応付けについて述べる．それぞれを対応付ける事で，曲線形状と線積分経路を新たな媒介変数でまとめ

て表記することが可能となる．続いて，関数の計算結果から色彩の決定方法および，積分区間の制御によるエネルギー波の移動表現について述べる．

3.1 レンダリングの基本方針

本手法は阿部 [14] による手法を基にする．空間中のエネルギー分布を関数を用いて基底し，分布空間に向けて視線 (レイ) を飛ばす．その視線上のエネルギー量を積算することで，1 本の視線上のエネルギー量が決まる．エネルギー波を投影する平面に対して，平面上のピクセル毎に視線を飛ばして積分計算を行うことで，エネルギー波全体の形状を表示する手法である．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を基にした手法であり，ボリュームデータを関数に置き換えることでレンダリング前の素材準備といった手間を省きつつ，エネルギー波の発光表現を高速に描画可能である．

本基本方針は，スカラー場に対する線積分と見なす事が出来る．視線の開始地点を $\mathbf{V}_s$，$\mathbf{V}_s$ からピクセルを通り十分遠い位置を視線の終点 $\mathbf{V}_g$ とする．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 (1) で表す．

$$\mathbf{V}(t) = (\mathbf{V}_g - \mathbf{V}_s)t + \mathbf{V}_s \tag{1}$$

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算する事となる．

基本方針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．図 2 は直線 GS 上におけるエネルギー量 α を積分する様子の模式図である．

3.2 分布関数の媒介変数表記

次に，エネルギー波分布関数を媒介変数表記する．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，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 (2) で表す．

$$\mathbf{C}(s) = \begin{pmatrix} \gamma s \\ y(s) \\ z(s) \end{pmatrix} \tag{2}$$

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

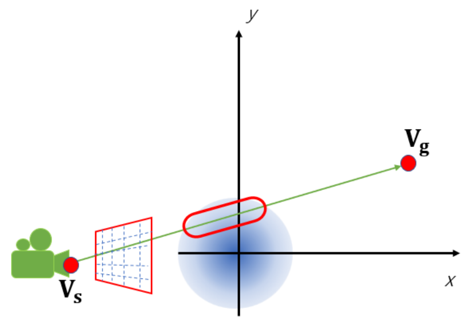


図 2: 基本方針の模式図.

降本論文では一般陽関数曲線と呼称する．

3.3 媒介変数同士の対応付け

本手法では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い，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) で算出する．

$$s = \mathbf{V}(t) \cdot \mathbf{D} \tag{3}$$

$t = 0$ 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 (4) とする．

$$\begin{cases} u_t = t_s + (t_g - t_s)u \\ u_s = s_s + (s_g - s_s)u \end{cases} \tag{4}$$

線積分の対象が u によって定まる．図 3 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図 3 においては曲面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 被積分関数

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 (5) とし

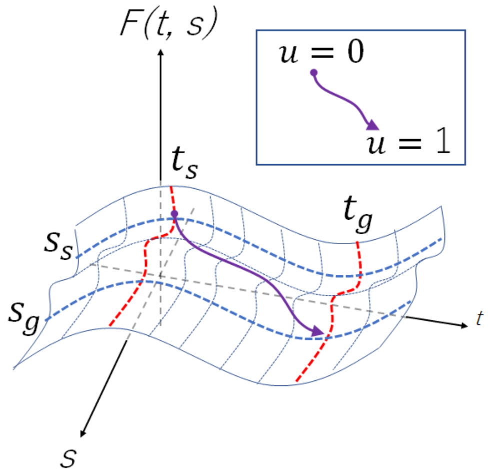


図 3: t,s の対応付を表す模式図

て基底する．

$$F(t,s) = F(u) = \frac{\alpha}{|\mathbf{V}(u) - \mathbf{C}(u)|} \tag{5}$$

$\mathbf{V}_g - \mathbf{V}_s = \mathbf{V}_r$，$t_g - t_s = t_r$，$s_g - s_s = s_r$ 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．

$$\begin{aligned}\mathbf{V}(u) &= \mathbf{V}_s + u_t\mathbf{V} \\ &= \mathbf{V}_s + t_s\mathbf{V}_r + ut_r\mathbf{V}_r \\ &= \begin{pmatrix} A+uB \\ C+uD \\ E+uF \end{pmatrix}\end{aligned} \tag{6}$$

$$\begin{aligned}\mathbf{C}(u) &= \begin{pmatrix} \gamma u_s \\ y(u_s) \\ z(u_s) \end{pmatrix} \\ &= \begin{pmatrix} \gamma s_s + \gamma s_r u \\ y(s_s) + y(s_r u) \\ z(s_s) + z(s_r u) \end{pmatrix} \\ &= \begin{pmatrix} I + Ju \\ K + y(Lu) \\ M + z(Nu) \end{pmatrix}\end{aligned} \tag{7}$$

さらに，

$$\begin{aligned}\beta &= \big((A-I) + u(B-J)\big)^2 \\ &+ \big((C-K) + uD + y(Lu)\big)^2 \\ &+ \big((E-M) + uF + z(Nu)\big)^2\end{aligned} \tag{8}$$

とすることで，

$$|\mathbf{V}(u) - \mathbf{C}(u)| = \sqrt{\beta} \tag{9}$$

となる．この式 (9) を式 (5) に代入し，一般陽関数とする．

3.5　提案関数の積分

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 4 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．

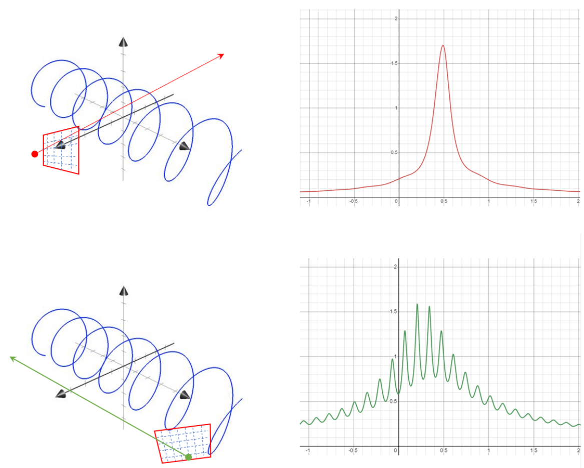


図 4: 各視線に応じたエネルギー量変化の様子

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量関数，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 [16] 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 [15] 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．そこで本手法では描画処理の高速化として自動積分法の適応型の考え [17] 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する大きさとする事で，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 5 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．

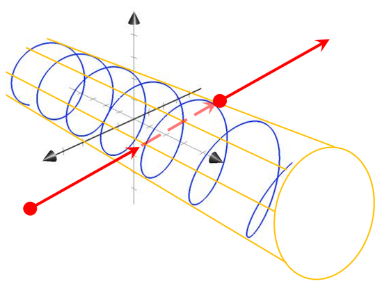


図 5: 螺旋を覆うバウンディングボックスと視線との交点

視線上の点 $\mathbf{V}(u)$ と螺旋上の点 $\mathbf{C}(u)$ は既に対応が取れている．この時螺旋の基準位置を $\mathbf{C}_b$，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 (10) を解く．

$$\begin{aligned} r + \epsilon &= |(\mathbf{V}_s + u_t\mathbf{V}_r) - (\mathbf{C}_b + u_s\mathbf{C}_r)| \\ &= |\,(t_r\mathbf{V}_r - s_r\mathbf{C}_r)\,u + (\mathbf{V}_s + t_s\mathbf{V}_r) - (\mathbf{C}_b + s_s\mathbf{C}_r)| \end{aligned} \tag{10}$$

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 6 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．

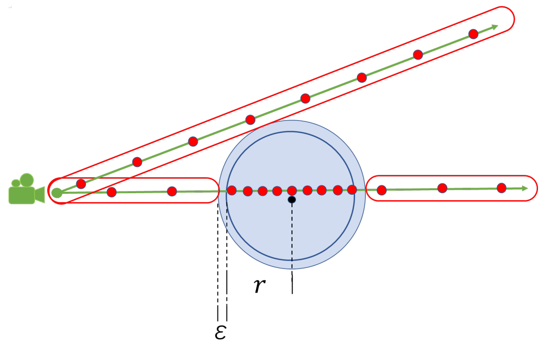


図 6: 円柱と視線の交差による分点数削減

3.6 色彩決定

本手法は式 (5) 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．表現し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 $0.0 \sim 1.0$ の範囲で規定しておく．それぞれの成分に対して式 (5) 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．エネルギー波の進行方向と並行な軸線上に近しい位置の場合，投影面上ではエネルギー波形状が密な状態となり，曲線の輪郭周辺のエネルギー量が増加する．結果として輪郭を覆う様に靄がかか

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．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，現実の光の特性を参考にし距離に応じてエネルギー波を減衰させる．投影面上では近くとも，分布空間中では遠い距離にあるエネルギー値に対して距離に応じた減衰補正を加えることで，極端な肥大化を防ぎつつ選択可能な表現の幅を広げる．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 (11) は本研究における距離に応じた減衰式である．

$$F(u) = \left(\frac{\alpha}{|\mathbf{V}(u) - \mathbf{C}(u)|}\right)(1.0 - u_t)^j \tag{11}$$

3.7　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 (12) となる．

$$F(u) = \int_0^k \frac{\alpha}{|\mathbf{V}(u) - \mathbf{C}(u)|}du + \int_k^1 \frac{\alpha}{|\mathbf{V}(u) - \mathbf{C}(k)|}du \tag{12}$$

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放出する様を表現可能である．線積分としての処理は最後まで続ける必要がある為，区間 $[k,1]$ の領域においては固定値 $\mathbf{C}(k)$ との距離を積分対象としている．

4　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 次ベジェ曲線形状
- B-Spline 曲線形状

についての適用を例示する．

4.1　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 (13) で表す．

$$\mathbf{C}(s) = \begin{pmatrix} \gamma s \\ r\sin(s\theta) \\ r\cos(s\theta) \end{pmatrix} + \mathbf{C}_b \tag{13}$$

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 (1)(4)(13) より $\mathbf{V}(u)$ と $\mathbf{C}(u)$ を以下のように表すものとする．

$$\begin{aligned} \mathbf{V}(u) &= \mathbf{V}_s + u_t\mathbf{V} \\ &= \mathbf{V}_s + t_s\mathbf{V}_r + ut_r\mathbf{V}_r \\ &= \begin{pmatrix} A + uB \\ C + uD \\ E + uF \end{pmatrix} \end{aligned} \tag{14}$$

$$\begin{aligned} \mathbf{C}(u) &= \mathbf{C}_b + \begin{pmatrix} \gamma u_s \\ r\sin(u_s\theta) \\ r\cos(u_s\theta) \end{pmatrix} \\ &= \begin{pmatrix} C_x^b + \gamma s_s + \gamma s_r u \\ C_y^b + r\sin(s_s\theta + s_r\theta u) \\ C_z^b + r\cos(s_s\theta + s_r\theta u) \end{pmatrix} \\ &= \begin{pmatrix} I + Ju \\ K + r\sin(G + uH) \\ L + r\cos(G + uH) \end{pmatrix} \end{aligned} \tag{15}$$

さらに，

$$\begin{aligned} \beta = &\big((A - I) + u(B - J)\big)^2 \\ &+ \big((C - K) + uD + r\sin(G + uH)\big)^2 \\ &+ \big((E - L) + uF + r\cos(G + uH)\big)^2 \end{aligned} \tag{16}$$

とすることで，

$$|\mathbf{V}(u) - \mathbf{C}(u)| = \sqrt{\beta} \tag{17}$$

となる．この式 (17) を式 (5) に代入し，一般陽関数による螺旋関数とする．

4.2 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 (18) で表す．

$$\mathbf{C}(s) = \begin{pmatrix} \gamma s \\ \sum_n (a_n \sin(ns\theta) + b_n \cos(ns\theta)) \\ \sum_n (c_n \sin(ns\theta) + d_n \cos(ns\theta)) \end{pmatrix} \tag{18}$$

以降は式 (7) と同様に媒介変数 u に置き換え，式 (5) となる様に u についてまとめる．

4.3 n 次ベジェ曲線形状

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 [15] では 2 次ベジェ曲線までの表現が限界だったのに対して，本手法では 3 次以上のベジェ曲線 [18] にも適用可能である．ベジェ曲線は制御点列 $\{\mathbf{P}_0, \mathbf{P}_1 \ldots \mathbf{P}_n\}$ によって式 (20) で定まる．

$$B_i^n(s) = {}_nC_i s^i (1-s)^{n-i} \tag{19}$$

$$\mathbf{C}(s) = \sum_{i=0}^{n} B_i^n(s)\mathbf{P}_i \qquad (0 \leqq s \leqq 1) \tag{20}$$

式 (2) 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 (7) と同様に媒介変数 u に置き換え，式 (5) となる様に u についてまとめる．

4.4 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 [18] 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 $\{\mathbf{P}_0,\ \mathbf{P}_1 \ldots \mathbf{P}_n\}$ と，位数 k およびノットベクトル $W = [x_0, x_1, \ldots, x_{n+k}]$ を用いて以下の式 (21) により曲線形状が決定できる．

$$\mathbf{C}(s) = \sum_{i=0}^{n} N_k^i(s)\mathbf{P}_i \qquad (2 \leqq k \leqq n+1) \tag{21}$$

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．

$$N_1^i(t) = \begin{cases} 1 & if \quad x_i \leqq t < x_{i+1} \\ 0 & if \quad t < x_i \quad or \quad t \geqq x_{i+1} \end{cases}$$

$$N_k^i(t) = \frac{(t - x_i)N_{k-1}^i(t)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - t)N_{k-1}^{i+1}(t)}{x_{i+k} - x_{i+1}} \tag{22}$$

式 (2) 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 (7) と同様に媒介変数 u に置き換え，式 (5) となる様に u についてまとめる．

5 検証と考察

本章では提案手法の検証結果について述べ，考察を行う．

5.1 検証環境

本手法を以下の表 1 のスペック PC にて検証を行った．

表 1: 検証用 PC の環境

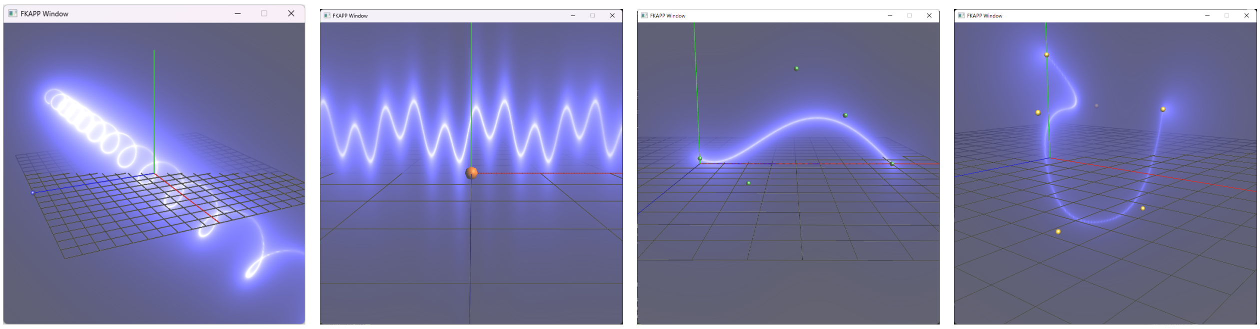
OS	Windows11 22H2
CPU	AMD Ryzen7 2700, 3.2GHz
Memory	16GB
GPU	NVIDIA GeForce RTX 2070, 8GB
Programming	OpenGL 4.1, GLSL 4.1

5.2 描画結果

提案手法によるプログラム実行画面を図 7 に示す．

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 8 はカメラ位置を変更した際の様子を示す．図 9 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 10 はベジェ曲線の制御点を変更した際の様子を示す．図 11 は B-Spline 曲線の制御点を変更した際の様子を示す．

次に色彩表現に関する描画結果を示す．図 12 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 13 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



(a) 螺旋曲線　(b) 三角関数和曲線　(c) ベジェ曲線　(d) B-Spline 曲線

図 7: 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．

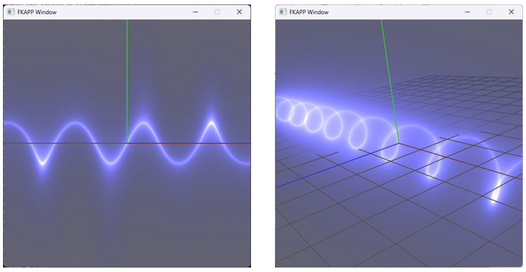


図 8: カメラ位置を変更した出力結果比較．

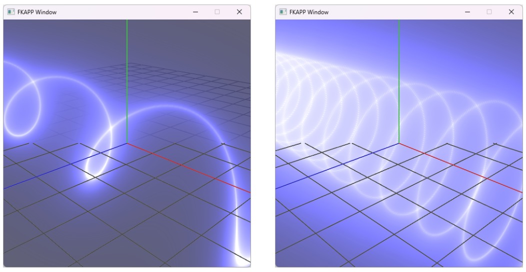


図 9: 調整係数 γ を変更した出力結果比較．

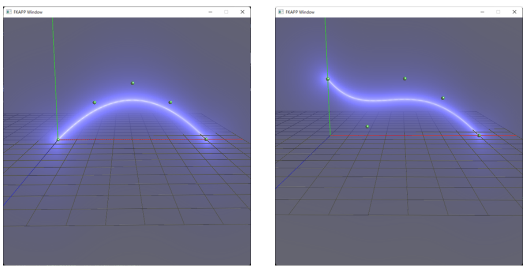


図 10: ベジェ曲線の制御点位置を変更した出力結果比較．

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．

次に積分区間の制御によるアニメーション表現に関する描画結果を示す．図 14 は時間変化に応じて積分区間を増加し，進行方向へエネルギー波が伸びていく様子を

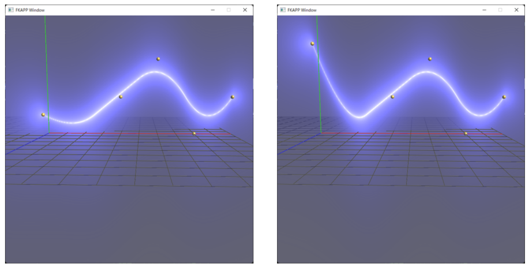
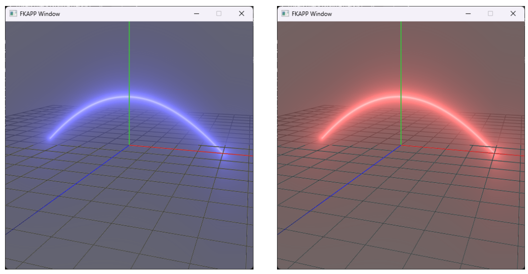


図 11: B-Spline 曲線の制御点位置を変更した出力結果比較．



(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 12: 基準色に応じた色変化の様子．

示す．

図 15 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任意となるが，提案関数の算出結果はこれらの補正計算に応じて見た目を変更可能である．異なるカメラ位置からの見え方を提示するが，両視点位置共に原点からは同距離にある．

図 16 は螺旋形状に補正を加えた際の描画結果である．

5.3　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 (Frame Per Second , FPS) とし，異なる解像度の投影

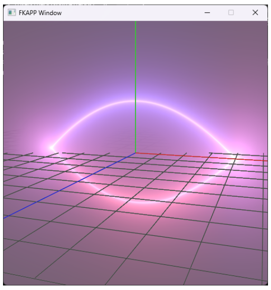
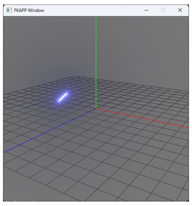
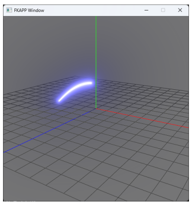


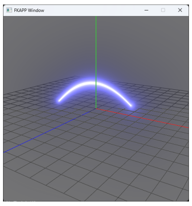
図 13: 異なる基準色曲線の計算結果合算の様子.



(a) $u_t : 0 \sim 0.1$

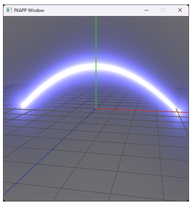


(b) $u_t : 0 \sim 0.4$

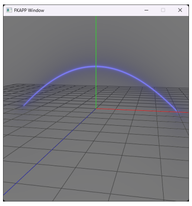


(c) $u_t : 0 \sim 0.8$

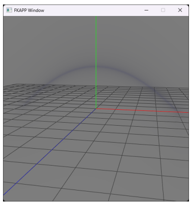
図 14: 時間変化に応じた積分区間変更によるエネルギー波の伸長表現.



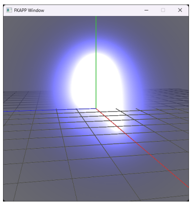
(a) $j = 0.0$, 視点位置 A



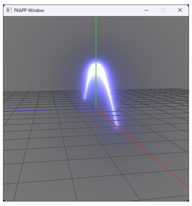
(b) $j = 1.0$, 視点位置 A



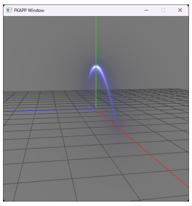
(c) $j = 2.0$, 視点位置 A



(d) $j = 0.0$, 視点位置 B



(e) $j = 1.0$, 視点位置 B

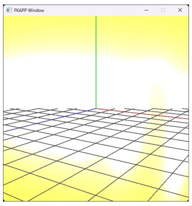


(f) $j = 2.0$, 視点位置 B

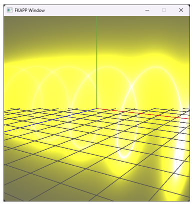
図 15: 減衰式を適用した曲線表示例. (a),(d) の $j = 0.0$ は減衰無しとなる. 減衰のパラメータ j 以外は (a) ～ (f) まで全て同値である.

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. 表 2 は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.

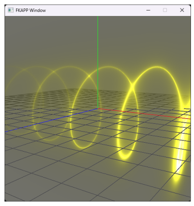
三角関数ベースの曲線 2 つについて, 高解像度かつ分点数増加に応じた処理速度低下が確認できたが, 一般的なリアルタイムレンダリングが必要とする 60FPS 程度



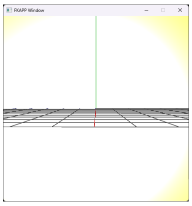
(a) $j = 0.0$, 視点位置 A



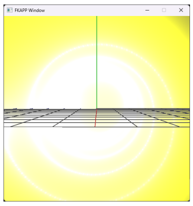
(b) $j = 1.0$, 視点位置 A



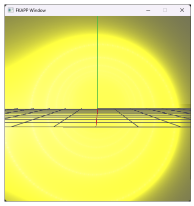
(c) $j = 2.0$, 視点位置 A



(d) $j = 0.0$, 視点位置 B



(e) $j = 1.0$, 視点位置 B



(f) $j = 2.0$, 視点位置 B

図 16: 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.

は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思われる. 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なったと思われる.

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 3 に示す. 表中の m と n はそれぞれ, 第 3.5 節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 1024 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

表 2: 実行速度 (FPS)

螺旋		
分点数 m	512*512	1024*1024
100	355.6	349.8
300	357.9	320.7
500	357.0	195.5
三角関数和		
分点数 m	512*512	1024*1024
100	343.5	335.4
300	344.3	254.3
500	345.5	165.1
4 次ベジェ		
分点数 m	512*512	1024*1024
100	358.8	267.0
300	351.2	94.0
500	219.2	57.0
B-Spline		
分点数 m	512*512	1024*1024
100	262.1	49.0
300	89.7	17.0
500	54.0	10.0

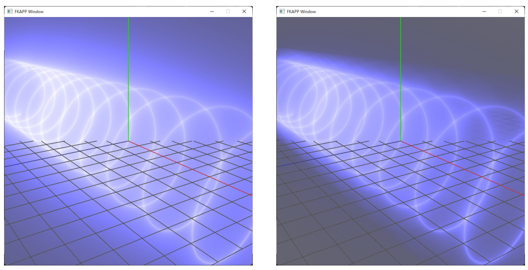
表 3: 分点数削減を伴う実行速度 (FPS)

螺旋				
m	n	2 点交差視線	FPS(削減前→後)	
100	2	5 割程度	349.8	→ 354.0
	4			→ 350.0
	2	全域		→ 350.7
	4			→ 349.3
500	2	5 割程度	195.5	→ 353.7
	4			→ 350.5
	2	全域		→ 226.0
	4			→ 227.1
B-Spline				
m	n	2 点交差視線	FPS(削減前→後)	
100	2	5 割程度	49.0	→ 116.0
	4			→ 192.0
	2	全域		→ 101.0
	4			→ 135.5
500	2	5 割程度	10.0	→ 26.0
	4			→ 43.0
	2	全域		→ 23.0
	4			→ 31.0

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．

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っ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 17 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 18 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 19 はカメラ位置を近辺に置き，分点削減の有無による描画結果を比較した画像である．



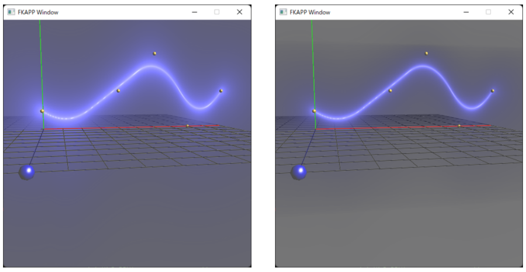
(a) 分点削減：未適用　(b) 分点削減：適用後

図 17: 遠方カメラ配置 - 分点削減有無による螺旋曲線比較

5.4 表現例の比較

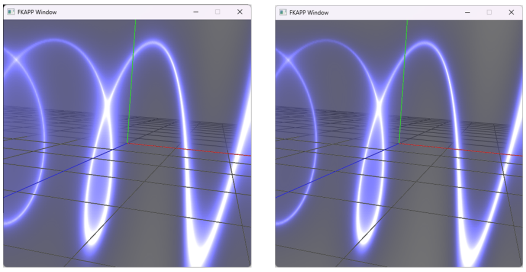
本手法で作成例と既存のコンテンツ事例とを比較する．螺旋曲線は主に，ドリルの様な螺旋形状を放出することで敵を穿つさまを示したり，竜巻の様な形状や模様で利用される．図 20 はアニメコンテンツ内で螺旋形状が象徴的な攻撃手法と本手法での作例比較である．螺旋形状と直線形状の組み合わせとなり，直線形状はベジェ曲線を用いている．

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出来るが，手数の多さを演出



(a) 分点削減：未適用　(b) 分点削減：適用後

図 18: 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 19: 近辺カメラ位置 - 分点削減有無による結果比較

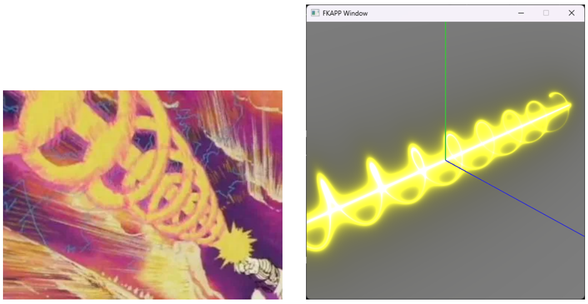


図 20: アニメコンテンツおよび本手法との作例比較 - 螺旋曲線．各種調整係数は螺旋曲線に対し，$\alpha = 35.0, r = 5.0, \gamma = 4.0, j = 1.0$ とし，ベジェ曲線に対し，$\alpha = 30.0, j = 1.0$ とした．

出典”ドラゴンボール”

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ェ曲線や B-Spline 曲線はこうした曲線形状，曲線軌道に適している．図 21 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．調整係数を同一設定した複数本の 4 次ベジェ曲線で構成し，積分区間 k についてそれぞれ異なる時間変化設定を施している．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．

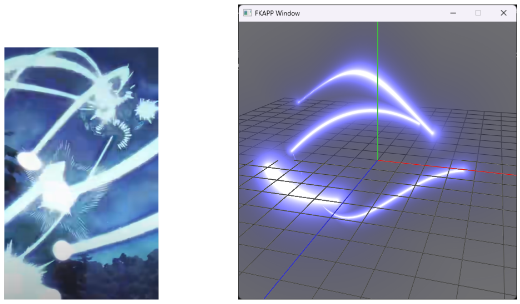


図 21: アニメコンテンツおよび本手法との作例比較 - 制御点による曲線．各種調整係数は $\alpha = 30.0, j = 1.0$ とした．

出典”葬送のフリーレン”

5.5　課題

実装内容より現行の課題点について述べる．リアルタイムな処理速度で螺旋形状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．プリミティブな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積分方法も併せて提案した．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言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．

本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。本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。

6 まとめ

本研究は関数を用いたリアルタイムエネルギー波形状表現の拡張を目的とし，既存手法ではリアルタイム描画が不可能であった曲線種に着目した．対象曲線の特定軸方向を線形単調増加とする事で，線積分中の媒介変数と曲線中の媒介変数同士を一対の組みとみなし，新たな媒介変数で線積分と曲線を結びつける手法を提案した．提案手法が複数の曲線種に適用可能な事例として，螺旋曲線 · 三角関数和曲線 ·3 次以上のベジェ曲線 ·B-Spline 曲線への適用を試みた．

提案手法を実装検証し，処理速度や出力結果について本目的が達成できた事を確認した．パラメータ制御により豊かな色彩表現や移動表現が可能である事を示し，既存コンテンツ表現例に近しい表現が本手法でも作成可能である事を確認した．本研究により，1 つの関数によるエネルギー波形状の大域的表現種類が増え，関数分布を用いたエネルギー波形状表現がより豊かになった．

今後は課題点を始めとして更なるエネルギー波形状の拡充や，時間を変数として関数に組み込み位置情報と時間情報によるエネルギー分布関数化によるアニメーション表現を追加検討していきたい．

参考文献

[1] 鳥山明. DRAGON BALL 大全集 - 鳥山全ワールド (7). 集英社, 1996.

[2] Unity. Visual effect graph. `https://unity.com/ja/visual-effect-graph`. 参照: 2022-3-16.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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=Qcjt3tyuNlU`. 参照: 2024-5-21.

[8] 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*, Vol. F-1, pp. 1 – 8, 2023.

[9] Z.PAN and D.MANOCHA. Efficient solver for spacetime control of smoke. *ACM Transactions on Graphics*, Vol. 36, No. 5, pp. 162:1 – 162:13, 2017.

[10] A.Stomakhin and A.Selle. Fluxed animated boundary method. *ACM Transactions on Graphics*, Vol. 36, No. 4, pp. 68:1 – 68:8, 2017.

[11] 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.

[12] J.Blinn. A generalization of algebraic surface drawing. *ACM Transactions of Computer Graphics*, Vol. 1, No. 3, pp. 235 – 256, 1982.

[13] 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 Com-*

munication Conference, Vol. 568, pp. 718 – 725, 1985.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] G.E.Farin. NURBS 射影幾何学から実務まで 第 2 版. 共立出版, 2001.

阿部　雅樹

2008 年東京工科大学メディア学部卒業．2010 年同大学大学院修士課程バイオ・情報メディア研究科メディアサイエンス専攻修了．2016 年より同大学メディア学部実験助手，現在に至る．コンピュータグラフィックスやゲーム制作に関する研究に従事．芸術科学会会員．

渡辺　大地

1994 年慶応義塾大学環境情報学部卒業．1996 年慶応義塾大学政策・メディア研究科修士課程修了．2016 年岩手大学工学研究科デザイン・メディア工学専攻博士後期課程修了．博士 (工学)．1999 年より東京工科大学メディア学部講師．2017 年より同准教授，2020 年より同教授，現在に至る．コンピュータグラフィックスやゲーム制作に関する研究に従事．芸術科学会，情報処理学会，画像電子学会，人工知能学会会員．