

2023年度 卒業論文

破片形状を考慮した物体の
リアルタイム破壊シミュレーションに関する研究

指導教員：渡辺 大地 教授

メディア学部 ゲームサイエンスプロジェクト

学籍番号 M0A20008

浅野 耀太

2024年1月

2023年度 卒業論文概要

論文題目

破片形状を考慮した物体の
リアルタイム破壊シミュレーションに関する研究

メディア学部

学籍番号：M0A20008

氏名

浅野 耀太

指導
教員

渡辺 大地 教授

キーワード

破壊、亀裂、物理シミュレーション、有限要素法、ビデオゲーム

ビデオゲームなどのインタラクティブなコンテンツにおいて破壊表現は多用される表現の1つであり、視覚的なリアリティや動的に反応を返す計算効率が求められる。リアルタイム破壊シミュレーションは物体にかかる外力を基に破片形状を決定する物理ベース手法と幾何学模様やアーティストによって破片形状を決める非物理ベース手法に分類できる。実際は非物理ベース手法が用いられるケースが多く、物理ベース手法がビデオゲームで用いられた事例は数少ないが、非物理ベース手法と比較してリアリティの高い破壊表現を行う事ができる。しかし、既存の物理ベース手法は生成される亀裂や破片形状がモデルの境界に依存しており、正確な亀裂や破片を生成することができないという問題がある。

そこで本研究ではより正確な亀裂や破片を生成できる物理ベース手法を提案する。本手法は有限要素法を使用し、破壊対象はボリュームメッシュで表現する。各要素にかかる主応力から破壊を行うか判定し、破壊する場合は破断面に沿って要素を分割することで物理的に正しい亀裂や破片を発生させることができる。また、リアルタイムで処理を行うために、各要素の分割可能回数を制限することで要素の増加量を抑制している。

本手法を上記で述べた既存の物理ベース手法手法と計算効率とリアリティの観点から検証を行った結果、既存の手法よりも物理的に正確で、より現実的な表現が可能となったが、計算効率の点では既存の手法よりも計算量が大きくなり、要素数の少ないモデルを使用するという条件付きでリアルタイムで実行することができる事が確認できた。

目次

第1章	はじめに	1
1.1	研究背景と目的	1
1.2	論文構成	4
第2章	提案手法	5
2.1	提案手法の流れ	5
2.2	応力の計算	6
	2.2.1 主応力の計算	7
	2.2.2 破壊の判定	7
2.3	各要素のリメッシュ処理	8
	2.3.1 四面体の分割	8
	2.3.2 分割可能回数の制限	11
	2.3.3 破断面の回転	12
	2.3.4 隣接リメッシュ処理	13
第3章	検証と考察	14
3.1	実装環境	14
3.2	検証用環境	14
3.3	検証方法	15
	3.3.1 比較手法	15
	3.3.2 要素分割の動作検証	15
	3.3.3 計算効率の検証	15
3.4	検証結果	16
3.5	考察	21
第4章	まとめ	23

謝辭 24

参考文献 25

目 次

2.1	x-y-z 座標系における応力テンソルの各成分の方向	6
2.2	要素分割の有無による亀裂の違いを表した模式図	9
2.3	四面体要素の分割の流れ	10
2.4	閾値による破断面の回転	12
3.1	検証で使用了モデル	16
3.2	指定した破断面と亀裂の発生点	17
3.3	Parker 手法での破壊結果	18
3.4	本手法での破壊結果	18
3.5	Parker 手法でのシミュレーション結果	19
3.6	本手法でのシミュレーション結果	20

第 1 章

はじめに

1.1 研究背景と目的

映画やビデオゲームといったエンターテインメントコンテンツにおいて、CG による破壊シミュレーションは頻繁に用いられる表現の一つである。例えば、ビルが倒壊するシーンや爆発によって壁が吹き飛ばされるシーン、陶器を落として粉々にする表現などが破壊シミュレーションによって再現される。しかし破壊シミュレーションに求められる特性は、映画とゲームでは大きく異なる。映画では物理的な正確さや、意図する演出が行えるかどうか最優先される一方、ゲームにおける破壊シミュレーションではリアルタイムで動作する計算効率が重視され、その上で物理的な正確さや視覚的なリアリティが求められる。また、ゲームにおける破壊表現は視覚的な演出のみではなく、仕掛けとしてユーザーが得る体験を向上させる役割も担っている。例として、対戦ゲームである THE FINALS[1] ではレベルに存在するオブジェクトの大半が破壊可能となっており、橋を破壊して対戦相手の進行を妨げたり、対戦相手がいる部屋にドアから侵入せずに天井を破壊して奇襲を仕掛けるなど、プレイヤーに自由な戦術を促すためのギミックとして破壊シミュレーションが用いられており、ゲーム体験の根幹を担っている。

リアルタイム破壊シミュレーションには様々な手法があり、ゲームに用いられる破壊手法を調査した Thomas ら [2] によると、リアルタイム破壊シミュレーションは主に事前破壊手法と、リアルタイム破壊手法の 2 種類に分けられると述べている。

事前破壊手法は、破壊対象となるオブジェクトに対して予め破片形状を定義しておき、オブジェクトに破壊が発生するほどの衝撃が加わった時に定義した破片形状へ分解する手法である。破壊時に物理的な計算やオブジェクトの分割を行わないため、計算効率の点で優れている。そのため現在最も多く使用されている手法であり、ゲームエンジンである UnrealEngine[3] にもその機能が組み込まれている。破片形状を作成する方法には、アーティストが手動で分割を行う方法や、幾

何学模様であるポロノイ図に沿って分割を行う手法 [4] 等がある。この手法を採用している作品として、シューターゲームである Battlefield シリーズや、アクションゲームである CONTROL[5] が挙げられる。Battlefield3[6] では、建物が破壊される時、部分的に破壊されたメッシュと交換し、同時に粉塵などのエフェクトを発生させることである程度のリアリティを確保しつつ、少ない計算量で破壊表現を実現している。また、関連研究として Sellan ら [7] は物体の幾何学的に脆い部分を考慮した破片形状を定義する手法を提案しており、視覚的なリアリティの高い破片を生成することができるが、同じ物体に対して繰り返し破壊を行うことができないという欠点がある。

リアルタイム破壊手法は、対象のオブジェクトに破壊が発生した時、定義された破壊パターンや、物体にかかる応力に従ってオブジェクトを分割する手法である。事前破壊手法との主な違いは、破片形状を事前に定義せず実行時に決めて分割を行うという点であり、計算コストは高くなるが破壊表現のリアリティも向上している。破片形状や破壊パターンの決め方によって、リアルタイム破壊手法はさらに物理ベース手法と非物理ベース手法に分けることができる。それぞれの手法を関連研究と使用例を交えて説明する。

まず、非物理ベース手法はアーティストが用意した破壊パターンや、動的に生成した破壊パターンに従ってオブジェクトを分割する手法である。一般的にはポロノイ図を破壊パターンとして使用することが多く、関連する研究として Müller ら [8] が提案した手法では衝撃点を中心にオブジェクトにポロノイ図を投影し、模様の境界線に従って分割を行う。非物理ベース手法の使用例としては対戦ゲームである Tom Clancy's Rainbow Six Siege [9] で用いられており、弾丸や爆発物によって壁や床に穴を開ける際に使用されている。穴を開ける際、事前に破片形状が決められていると意図した形状の穴を開けられない可能性があるため、この手法を採用したのだと推察できる。実際に、このゲームの制作に携わった Julien[10] によると、破壊パターンはゲーム中の入力とオブジェクトの物質パラメータに基づいて生成されると述べていた。非物理ベース手法は意図した形状の穴や破片を作り出したい場合に適しているが、予め用意した破壊パターンを用いるため、発生する亀裂の方向や破片形状が物理的に正確ではない。

次に、物理ベース手法はオブジェクトが衝撃を受けた際、物体内部の各点にかかる応力を計算し、その大きさと方向に従って破壊を行う手法である。関連研究として Parker ら [11] の手法がある。この手法は四面体有限要素法を用いて物体の変形と応力の算出を行うため、破壊対象のオ

ブジェクトは四面体形状の要素を組み合わせたボリウムメッシュで表され、各要素ごとに応力を計算する。最終的にかかる各節点の応力の大きさによって破壊を判定し、破壊する際は応力の垂直方向に最も近い四面体同士の境界面を亀裂として分離する。このプロセスによって物体の弾性変形と破壊表現を同時に行うことができる。これに類似する手法として、破壊対象を剛性の高い物質に限定することで計算効率を向上させた Müller ら [12] の破壊手法や、弾性体が剥離する表現をリアルタイムで実現した大嶋ら [13] の手法が存在する。Parker ら [11] の手法はアクションゲームである Star Wars: The Force Unleashed[14] で用いられており、破壊が発生する前に変形が起きやすい木材や一部の金属の破壊表現に用いられている。この手法は使用する用途によっては最もリアリティの高い破壊表現を行うことができるが、物理的な計算を多く行うため計算量も事前破壊手法や非物理ベース手法と比較して最も大きい手法になっている。しかし、リアリティの点で欠点が存在しており、破壊を行う際生成する亀裂の進行方向を既存の四面体の境界から選択するため、結果的に破片形状が人工的な見た目になる可能性がある。この問題に対して Parker ら [11] は各要素にグラフィックオブジェクトを埋め込み視覚的なリアリティを補うことで対処していたが、応力方向に忠実ではない亀裂や破片を生成する可能性は依然として存在する。

本研究の目的は、リアルタイムで実行でき、既存の破壊手法よりも現実的な亀裂や破片を生成する手法を実現することである。この目的が達成されることによって、ゲームなどのインタラクティブなコンテンツにおいてよりリアリティのある破壊表現を行うことができる。提案手法は Parker ら [11] の手法をベースとしており、破壊対象はボリウムメッシュで表現し、有限要素法によって物体内部の応力を計算する。各要素の節点にかかる引張応力から破壊を行うか判定し、破壊する場合、上記で述べた問題点に対処するため破断面に従って要素を分割することで既存の四面体要素の境界に依存せずに物理的に正しい亀裂を形成し、結果的に現実的な形状の破片を生成できる。

本手法の検証を Parker ら [11] の手法と比較して行った結果、物理的に正確な亀裂を形成することによってより現実的な破壊表現を行うことが可能となりリアリティが向上した。しかし、計算効率の観点からは要素を分割することによって要素数が増加するためその分計算量が増加するが、破壊対象となるモデルの要素数によってはリアルタイムで動作することが確認できた。

1.2 論文構成

本論文は全 4 章で構成する。第 2 章では本研究の提案手法の説明を行う。第 3 章では提案手法をリアリティと計算効率の観点から検証を行い、考察を述べる。第 4 章では研究のまとめを述べる。

第 2 章

提案手法

本章では本研究での提案手法を説明する。本研究の提案手法は O'Brien ら [15] の有限要素法を用いた破壊モデリング手法 (以下 O'Brien 手法) と、O'Brien ら [15] の手法をビデオゲーム向けに改良した Parker ら [11] のリアルタイム破壊手法 (以下 Parker 手法) をベースとしており、この 2 つの手法の間中間的な手法となっている。以下ではまず提案手法の全体的な流れを説明し、その後提案手法を理解する上で重要な部分である応力計算や各要素のリメッシュ処理について説明する。

2.1 提案手法の流れ

本研究の提案手法は四面体要素を用いた有限要素法に基づいており、四面体で構成されたボリュームメッシュを対象に破壊を実行できる。破壊アルゴリズムの流れとしては以下の手順となる。

1. 各要素にかかる主応力の算出
2. 分割対象となる要素のリメッシュ処理

破断する要素を特定するためには、まず変形したボリュームメッシュの各要素にかかる応力を計算する必要がある。そして計算した応力を主応力に変換し、主応力の正の値である引張応力から各要素の破壊の判定を行い、対象となった要素を破断面に従って分割する。手順 2 については提案手法として新規性を含む部分になっており、2.3 節で説明する工夫点によって、各要素の増加を抑制している。上記の破壊アルゴリズム以外にも、Parker 手法ではビデオゲーム向けに衝突処理等にも工夫が施されているが、本研究は破壊時のモデリングに焦点を当てているためここでは割愛し、各要素のリメッシュ処理を中心に説明する。

2.2 応力の計算

応力とは、ある物体に力を加えた際に反発力として物体の内部に内力が発生するとき、物体内部の任意の面にかかる単位面積あたりの内力を示すものである。応力はさらに、面の法線方向にかかる垂直応力と、面の接線方向にかかるせん断応力に分解することができる。物体内のある点にかかる応力を表す場合、その点を通る3つの直交する面を基準として表すことが一般的であり、3つの直交面にかかる垂直応力とせん断応力を1つのテンソルで表す。例として直交座標系 $x-y-z$ において応力テンソル σ は式 (2.1) のように表すことができる。

$$\sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} \quad (2.1)$$

ここで応力テンソル σ の各成分は該当する応力の大きさを表し、応力の方向は図 2.1[16] で表されている方向と一致する。

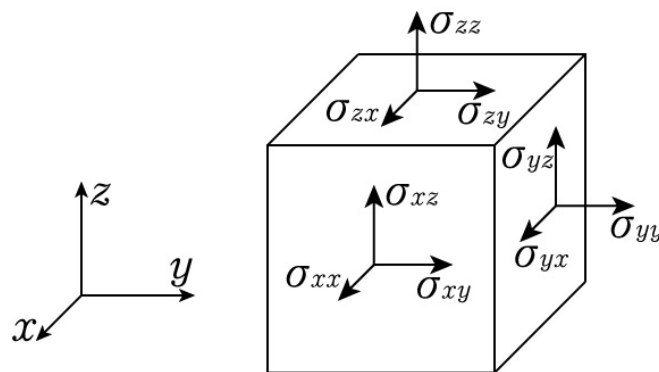


図 2.1 $x-y-z$ 座標系における応力テンソルの各成分の方向

<https://physics-school.com/stress-tensor/>

応力テンソルは 3×3 の対象行列で表す。各成分の方向は、ある成分を σ_{ij} とすると、 i 軸方向を法線とする面に対してかかる j 方向の力という見方になる。つまり、応力テンソルの対角成分は座標系の各軸方向にかかる垂直応力を表す。

有限要素法においては、各要素にかかる応力を1つの応力テンソルで表し、要素内の応力は一様であるものとして計算する。

2.2.1 主応力の計算

有限要素法において、外力を受ける物体が破断するか、また破断する場合物体内のどの点で破断するか求めたい場合、各要素における主応力を求める必要がある。主応力とは、応力テンソルのせん断成分が全て0になる座標系をとったときの応力を示す。このときの座標系を主軸座標系と呼ぶ。主軸座標系で表す応力テンソルは対角行列であり、各成分の方向は全て主軸座標系の各軸の方向に対応する。任意の座標系で表した応力テンソルを対角化すると、主軸座標系で表した応力テンソルとなる。例として主軸座標系 $x' - y' - z'$ における主応力 σ' を式 (2.2) のように表す。

$$\sigma' = \begin{bmatrix} \sigma'_{x'x'} & 0 & 0 \\ 0 & \sigma'_{y'y'} & 0 \\ 0 & 0 & \sigma'_{z'z'} \end{bmatrix} \quad (2.2)$$

ここで、 σ' の各対角成分は値の大きいものから順に最大主応力、中間主応力、最小主応力と呼ぶ。これらは正の値の場合は引張応力を表し、負の値の場合は圧縮応力を表す。

このように、応力テンソルを対角化することで要素にかかる応力を単純な引張応力と圧縮応力に分解することができる。主応力を求めることは一般的に物体が受ける負荷を評価するために行われることが多く、本手法では各四面体要素が受ける負荷によってその要素が破断するか判定するために用いる。

2.2.2 破壊の判定

まず、前提として1つの四面体要素は4つの節点からなり、各節点は四面体要素同士を繋ぐ役割を担っている。各要素の破壊の判定方法は、その要素にかかる最大主応力と、物質ごとに設定するパラメータである破壊閾値を比較し、最大主応力が上回る場合破壊する。亀裂の発生点となる位置については要素の重心点ではなく、要素が持つ4つの節点のうち、亀裂の先端としてマークされた節点からランダムに選択する。マークされた節点が無い場合は4つの節点からランダムに選択する。こうする理由は2つあり、第一に新しい亀裂が発生するより、既存の亀裂から伝播

する可能性が高いためであり、第二に、実際の物質には無数の微視的な傷が存在し、その傷から亀裂が発生する可能性が高いためである。亀裂の先端となる節点については、2.3 節の 2.3.1 項で説明する。

この破壊の判定方法は、弾塑性物体の破壊をリアルタイムでシミュレーションする方法を提案した Müller ら [17] の手法に基づいており、高速な処理に適している。この方法以外にも O'Brien ら [15] の手法から考案された、各節点に作用する力を示すテンソルである、分離テンソルを計算して物質の靱性パラメータと比較することで破壊を判定する方法があるが、この方法はまず各要素に作用する主応力を 4 つの節点に作用する力として分解し、最終的に各節点にかかる引張応力と圧縮応力の釣り合いを計算する必要があるため、計算コストが Müller ら [17] の判定手法より高くなる。

2.3 各要素のリメッシュ処理

各要素のリメッシュ処理の流れとしては、まずある要素の破壊が発生し、亀裂の発生点となる節点が決められたとき、破断面を定義する必要がある。破断面は最大主応力の方向に垂直であり、亀裂の発生点となる節点を通る。次に亀裂の発生点となる節点を 2 つに分割し、元の節点に接続されている全ての四面体要素を破断面と交差判定を行う。このとき交差している要素は破断面に沿って分割する。その後それらの要素を分割された 2 つの節点にそれぞれ再度割当て、結果として物体に破断面に沿った亀裂が形成できる。

このリメッシュ処理は O'Brien 手法に基づいたものであるが、本提案手法では破壊によってシーン内の四面体が無制限に増加し、リアルタイムでの動作が困難になることを防ぐために、各要素の分割回数に制限を設けている。以降の項ではこの変更点を踏まえつつ、リメッシュ処理の詳細を段階に分けて説明する。

2.3.1 四面体の分割

破断面が定義されると、物体に破断面に沿った亀裂を形成する必要がある。Parker 手法では既存の各要素の境界を亀裂とすることで、四面体を分割することなく破壊を行うが、亀裂の進行方向が既存の各要素の境界に依存するため生成する破片の形状が人工的になってしまう可能性がある

る。そのため本手法では O'Brien 手法と同様に四面体を破断面に沿って分割することで、正確な亀裂を形成する方法を選択した。この違いを模式的に表したものを図 2.2 に表す。

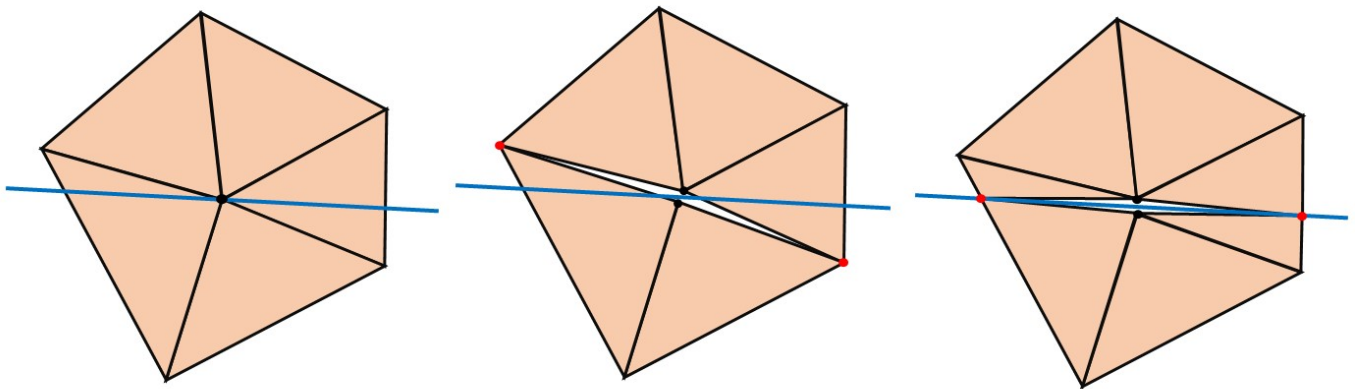


図 2.2 要素分割の有無による亀裂の違いを表した模式図

3次元の四面体要素を平面として捉えており、各要素は三角形で表している。左図が亀裂を形成する前の様子を表しており、青色の線分は破断面を表し、黒点は亀裂が発生する節点を表している。Parker 手法のように、既存の各要素の境界を亀裂とした場合は図の中央のようになり、破断面に最も近い節点が亀裂の先端となる。一方で O'Brien 手法のように要素を破断面に沿って分割する場合は右図のようになり、破断面と交差する辺上に節点を生成し亀裂の先端とするため、破断面に正確な亀裂を形成することができる。

分割の手順としては、まず亀裂の発生点となる節点を2つに分割し、これらを n^+ 、 n^- とする。ここで n^+ は破断面の法線側、すなわち最大主応力の方向側に位置し、 n^- はその裏側に位置するものとするが、この時点では2つの節点は同じ位置に存在し、同じ速度を持つ。

次に、元の節点と繋がっているすべての四面体要素と破断面が交差しているか判定し、交差していない場合はその要素が破断面のどちら側に位置しているかによって n^+ 、または n^- に再割り

当てする。要素が破断面と交差している場合、破断面に沿って分割する。メッシュの整合性を保つために全ての要素は四面体である必要があるため、基本的に交差する要素は3つの四面体に分割される。1つの要素を分割する流れを図2.3に表す。

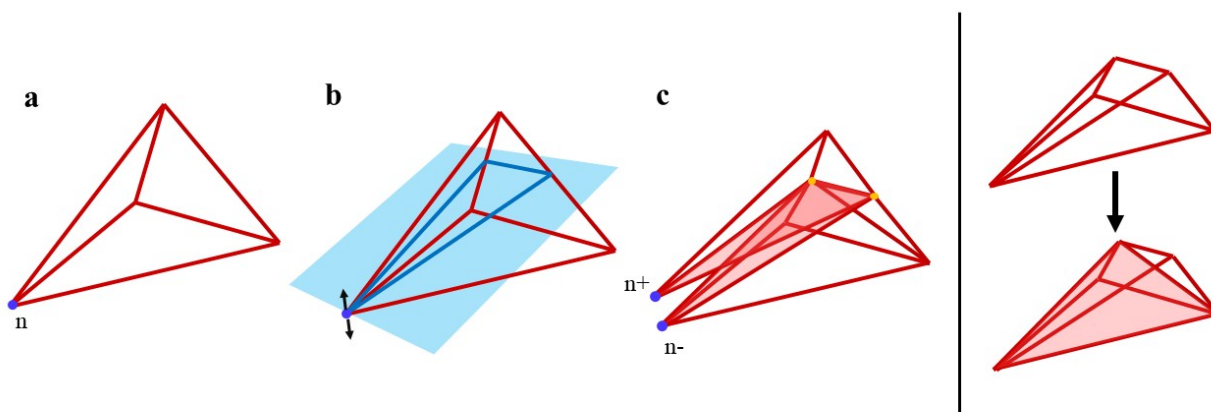


図 2.3 四面体要素の分割の流れ

図 2.3 では左図からアルファベット順に分割の段階を示している。破断面に沿って要素の分割を行う際、四面体が持つ2つの辺と破断面が交差し、それらの交差点に新たな節点を生成する。ここで、生成した2つの節点は分割した要素においてのみ存在するものであり、分割した要素に隣接する各要素には反映されないため、2つの節点にそれぞれ最も近い既存の節点を亀裂の先端としてマークしておく。

この時点で平面の片側に1つの四面体が形成され、もう一方には5つの節点をもつ五面体が形成されるため、整合性を保つために五面体を2つの四面体に分割する。ここで注意すべきことは、亀裂として切り離すのは破断面が交差している箇所であり、五面体から分割された2つの要素は接続されていることである。

2.3.2 分割可能回数の制限

2.3.1 項で述べたように、亀裂の発生点となる節点と隣接し、破断面と交差する要素は3つの四面体に分割する。分割された要素に強い応力が作用すれば、さらに3つの要素に分割することができる。ここで問題となるのは、破壊を繰り返すことで四面体要素が無制限に増加し、フレームレートを維持することが困難になることである。

この問題に対して Parker 手法では要素の分割を行わないことで、要素の増加を防ぎ、リアルタイムでの動作を実現していた。しかし要素の分割を行わないことで形成される亀裂は破断面に正確に沿ったものではなく、既存の要素の境界から近似したものになるため、物体の体積に対して存在する要素の密度が低いほど生成される破片の形状が人工的になってしまう。また、Koschier ら [18] は物体の応力が高い領域のみ可逆的な四面体分割を行うことで、亀裂が入りやすい部分のみ要素の密度を高めて処理の高速化を実現したが、この手法はリアルタイムでの実行を想定していない上に、物体を粉々になるまで破壊した際にはモデルの全領域の要素密度が高くなってしまう。

そこで本手法では初期状態のモデルに対して各要素の分割可能回数を制限することで、ある程度まで正確な亀裂を形成することができ、かつリアルタイムでの処理を維持できる手法を提案する。例として、要素数 100 のモデルに対して各要素の分割可能回数を 1 回とし、全ての要素を限界まで破壊すると最も分割回数が増える場合で要素数は 300 になる。実行する環境や最適化の有無によってフレームレートを維持できるかどうかは結果が分かれる所ではあるが、少なくとも要素の増加量は計画でき、ユーザーはその増加量を踏まえた上で制作を行うことができる。

生成される破片形状や亀裂のリアリティについては、各要素において分割回数が制限されるまでは破断面に沿った正確な亀裂を形成することができるが、制限によって分割できない場合は Parker 手法と同様に既存の要素の境界から近似した亀裂を形成することになる。但し、モデルに存在する要素が偏りなく並んでいる場合、1 回以上分割した要素は分割していない要素と比較して小さくなっているため、正確な破断面に近い亀裂を形成できる可能性が高くなる。

2.3.3 破断面の回転

四面体要素を分割すると一般的に3つの要素に分けられるが、例外として、2つの要素に分割できる場合がある。これは破断面がちょうど亀裂の発生点以外の節点を通る場合である。また、破断面がちょうど既存の要素の境界と一致する場合は、要素を分割することなく亀裂を形成できる。これらのケースは要素の増加を抑えるが、ほとんど起こり得ないことである。

そこで、本手法は O'Brien 手法に基づき、2種類の閾値を用意した。1つは破断面と既存の節点間の距離で、もう1つは既存の節点から破断面が要素の辺で交差する点がなす線と破断面との間の角度である。これらのいずれかが満たされない場合、破断面を既存の節点を通るように回転させる。こうすることで上記で述べた2つのケースを作り出し、要素の増加を抑えることができる。1つは破断面と既存の節点間の距離を x 、節点から破断面が要素の辺で交差する点がなす線と破断面との間の角度を θ として、閾値によって破断面を回転する流れを表した模式図を図 2.4 に示す。

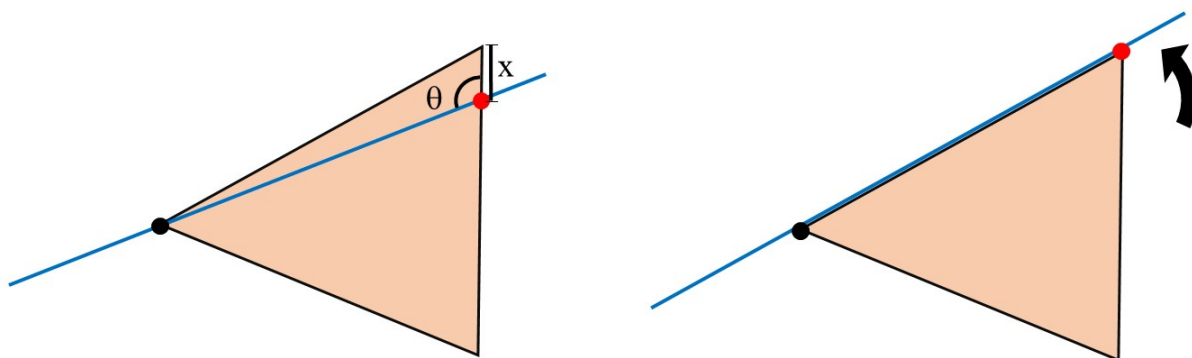


図 2.4 閾値による破断面の回転

図 2.2 と同様に、三角形は要素、破断面は青色の線分、赤点は亀裂の先端、黒点は亀裂が発生する節点を表している。

破断面をある程度回転させることを許容する理由は2つあり、1つは閾値によって破断面を回転しない場合、分割によってほぼ平面に近い形状をした要素が生まれる場合があり、その要素は有限要素法のシステムに悪影響を及ぼし物体の挙動を不安定にする可能性があるためである。もう1つは、適切な閾値を設定した場合、破断面を少し回転させて近似した亀裂と、要素を分割して形成した亀裂に視覚的な差はあまり無く、要素数を2つ増やすことで今後の処理に負担をかけやすくするデメリットの方が大きいためである。

2.3.4 隣接リメッシュ処理

本手法のリメッシュ処理は、基本的に O'Brien 手法に基づいており、同じリアルタイムで実行することを目標とした Parker 手法のリメッシュ処理とは四面体要素を破断面に沿って分割するという点で異なっているが、本手法は分割による要素の増加を抑制するために O'Brien 手法のリメッシュ処理から一部変更を加えた点がある。それはある要素を分割した際に、その要素に隣接する要素をさらに分割するというプロセスを省略したことである。このプロセスは本来メッシュの一貫性を保つために行われるものであるが、隣接する全ての要素を3つの要素に分割するため、破壊する度に多くの要素が発生し、要素の増加量を予測することも困難になる。このプロセスを行わないことで発生する弊害として、亀裂の先端が隣接する各要素に反映されず、亀裂の伝播が正確に行われないことが挙げられる。しかし破壊による要素の最大増加量を予測できないことは、リアルタイムで実行することを目標とする本手法において致命的な欠陥になる。そのため隣接リメッシュ処理は行わず、要素の増加量を最小限に抑えることを優先した。

第 3 章

検証と考察

本章では、本研究で行った検証について実装環境や検証に用いた環境を踏まえた上で説明し、その結果について考察する。

3.1 実装環境

本手法の検証を行うために、オープンソースの物理エンジンである BulletPhysics(以下 Bullet)[19] と、リアルタイム 3DCG に対応している FineKernelToolKit(以下 FK)[20] を用いて本手法の実装を行った。実装方法としては、本手法に必要な四面体有限要素法は Bullet 内に含まれており、そのシステムに 2 章で述べた破壊アルゴリズムを組み込むことで破壊シミュレーションを実装し、その可視化のために FK を使用した。尚、検証では主にリメッシュ処理の観点から既存手法と比較するため、破壊シミュレーションには並列化などの最適化は施していない。

3.2 検証用環境

検証の際にシミュレーションを行った環境を表 3.1 に示す。

表 3.1 検証環境

CPU	Intel Core i7-11800H 2.3GHz
メモリ	16.0GB
GPU	NVIDIA GeForce RTX 3060 Laptop

また、全ての検証はリリースビルドで実行した結果を元に行った。

3.3 検証方法

本節では、検証を行う方法について述べる。本手法の特徴である要素分割の動作検証と、計算効率の検証を既存手法と比較して行った。

3.3.1 比較手法

本手法は O'Brien 手法と、Parker 手法がベースとなっており、これらの中間的な手法となっているが、検証では同じリアルタイムで実行することを目標とした Parker 手法と比較し、計算効率やリアリティの観点から検証を行う事とした。Parker 手法を再現する上での注意事項として、この手法はビデオゲーム向けの使用を目的としており、衝突処理等に計算効率を向上させるための工夫がなされているが、本検証では主に破壊によるリメッシュ処理の点で比較を行うため、それ以外の部分は実装していない。つまり各要素の分割の有無での比較になる。

3.3.2 要素分割の動作検証

本手法の特徴である、破断面に沿って要素を分割するという点で動作検証を行った。要素の分割を行わない Parker 手法と本手法で、それぞれ同じ節点に対して同じ破断面で破壊を行った際の結果を比較し、本手法では破断面に忠実な亀裂を生成できているか検証を行った。

また、一つの例として壁に球体を衝突させて破壊を行うシミュレーションをそれぞれの手法で行い、要素の増加量を比較した。

3.3.3 計算効率の検証

計算効率については、2種類の検証を行った。まず、1つ目は初期状態から同じ節点で破壊が発生した際のリメッシュ計算時間の比較を行った。次に、モデルを各要素の分割可能回数の限界まで粉々に破壊した際に1フレームにかかるすべての処理の計算時間を比較した。

3.4 検証結果

3.3 節の通りにそれぞれ検証を行った。全ての検証は図 3.1 に示す壁を模したボリウムメッシュに対して行った。このモデルは初期状態で要素数 50、節点数 27、辺の数が 101 となっており、各要素の配置の偏りは少ない。

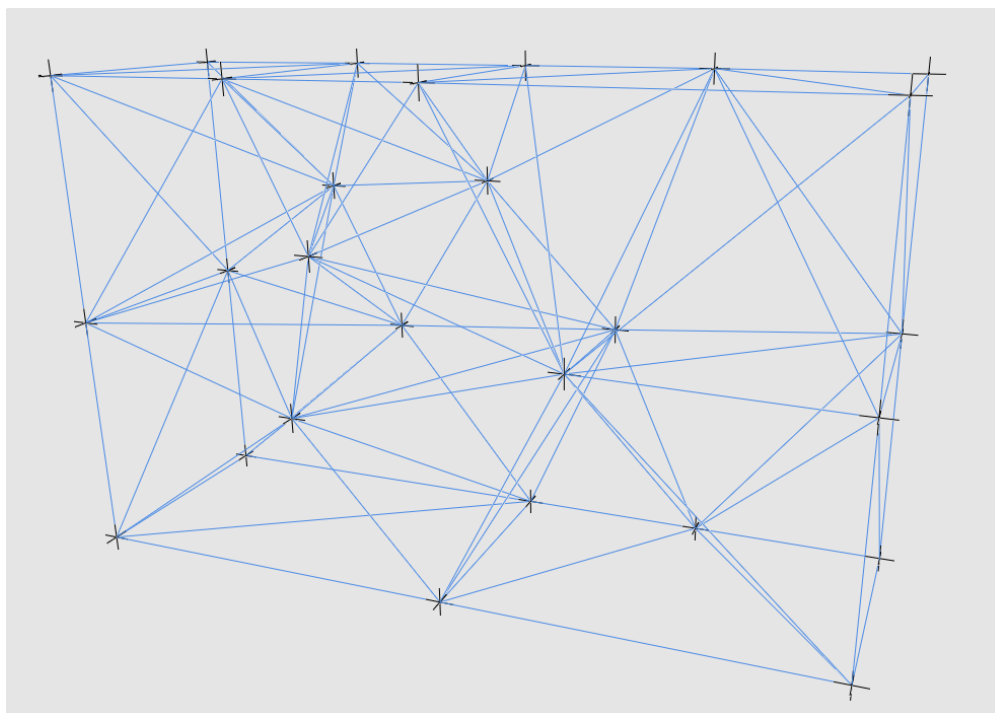


図 3.1 検証で使したモデル

モデルのワイヤフレームを可視化しており、辺は青色、節点は黒色の十字で表している。

3.3.2 項の説明の通り、一つの節点に対して破断面に従って破壊を行った。結果、本手法では破断面に忠実な亀裂を生成できることが確認できた。

破壊を行う前のモデルと入力した破断面、亀裂の発生点を表したものを図 3.2 に示す。また Parker 手法で破壊を行った結果を図 3.3 に示し、本手法で破壊を行った結果を図 3.4 に示す。破壊が発生し、亀裂の発生点となり分裂した節点は緑色で表し、そこから亀裂が伝播して亀裂の先端部分となった節点は赤色で表している。これらの実行結果は周辺の要素によって形が保たれているため実際に亀裂が発生したか確認し辛いですが、緑色の節点と赤色の節点に結ばれている辺が亀裂の境界と捉えることができる。Parker 手法の結果では、亀裂の発生点に隣接する要素を持つ各

節点のうち最も破断面に近い節点が亀裂の先端となっている。それに対して本手法の結果では亀裂の発生点に隣接する要素が破断面に沿って分割され、それによって生成された節点が亀裂の先端となっていることがわかる。

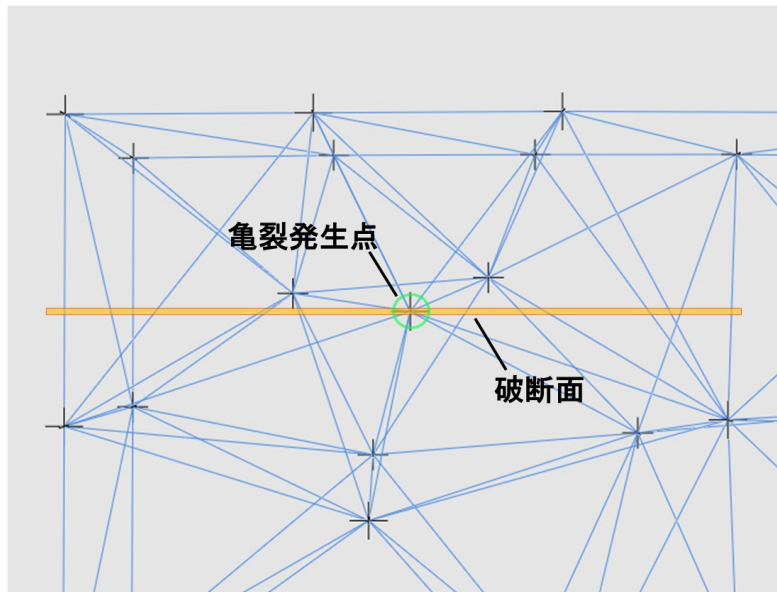


図 3.2 指定した破断面と亀裂の発生点

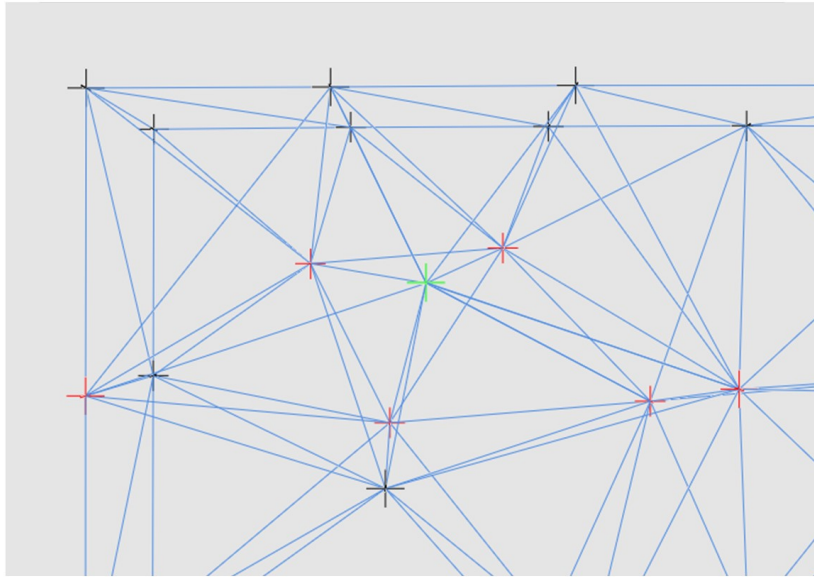


図 3.3 Parker 手法での破壊結果

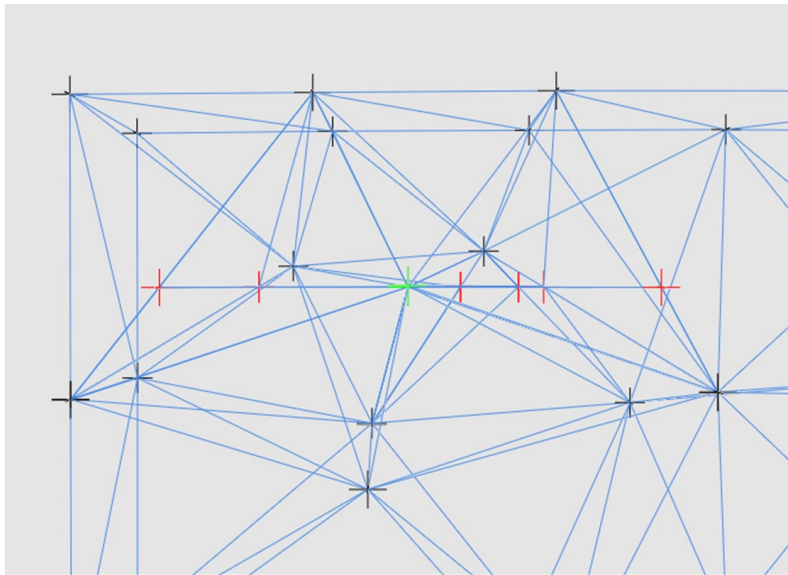


図 3.4 本手法での破壊結果

また、両手法での表現の違いを示すために、壁に球体を衝突させるシミュレーションを行った。Parker 手法での結果を図 3.5 に示し、本手法での結果を図 3.6 に示す。両手法での要素の破壊閾値は 560 に設定した。また本手法でのシミュレーションでは各要素の分割可能回数は 1 回に設定

し、破断面の回転を行う閾値は 0.8m と 0.5 ラジアンに設定した。

シミュレーションの結果、本手法は Parker 手法と比較して、より細かな破片を生成できることが確認できた。特に球体が衝突した周辺の応力が大きい領域は細かい破片が多く発生した。また、図 3.6 と図 3.5 の結果から、両手法で物質に与えるパラメータが同じでもシミュレーションの結果が大きく異なることが確認できた。

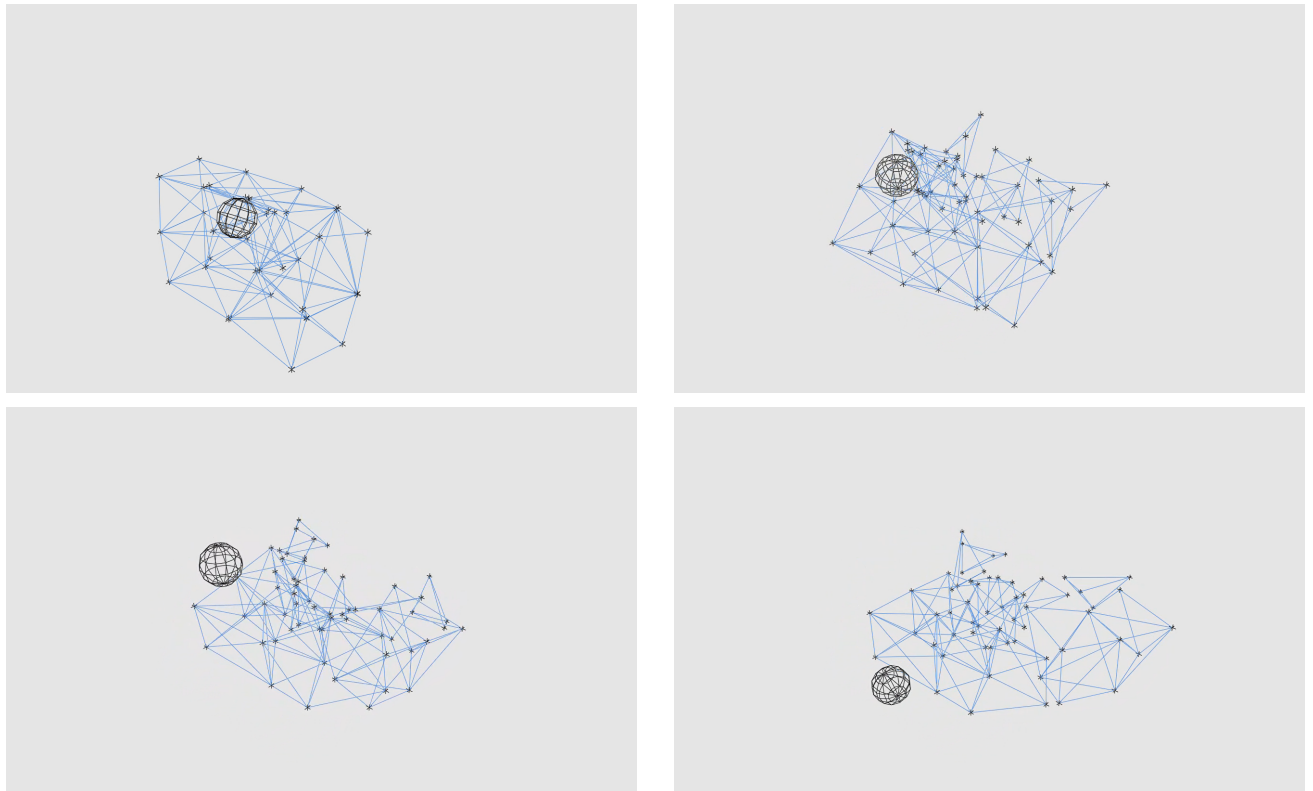


図 3.5 Parker 手法でのシミュレーション結果



図 3.6 本手法でのシミュレーション結果

次に、計算効率に関する検証結果について述べる。

まず両手法で同じ節点に対して破壊を行った際のリメッシュ処理にかかる時間を計測した。計測方法としては、亀裂の発生点とした節点に隣接する要素数が 13 で、一度に破断する要素数が比較的多いケースでリメッシュ処理を行い、計測の誤差を抑えるためにこの処理を 10000 回行った後、その平均値を 1 回のリメッシュ処理にかかる計算時間とした。結果は Parker 手法のリメッシュ処理時間は 485.2 マイクロ秒となり、本手法は 2569 マイクロ秒となった。本手法は各要素の分割を行う分計算時間が増えているが、リアルタイムで実行するには十分な計算時間であることが確認できた。

また、両手法においてモデルに対して破壊を繰り返し行い、可能な限り細分化した状態での 1 フレームの更新にかかる処理時間を計測した結果を表 3.2 に示す。本手法のシミュレーションでは各要素の分割可能回数は 1 回に設定し、破断面の回転を行う閾値は 0.8m と 0.5 ラジアンに設定した。

表 3.2 モデルを細分化した状態での処理時間 (μs)

	要素数	最大処理時間	最小処理時間	平均処理時間
Parker 手法	50	13000	7000	8946
本手法	125	61000	39000	48402

モデルの細分化を行った状態で、1000 フレームに渡って計測した結果を示している。Parker 手法では要素数が変わっておらず、本手法では 2.5 倍に増加している。要素の最大増加量は初期状態の要素数から 3 倍となっているが、破断面の回転により要素の増加量が抑制された結果を表している。1 フレームの計算時間に関しては両手法で大きな差が出ており、Parker 手法では 60fps 以上での動作が確認できるものの、本手法ではある程度破壊を行うまでスムーズに動作したが、要素数が 125 まで増加した状態ではリアルタイム処理の要件を達成することはできなかった。

3.5 考察

要素分割の動作検証の結果からは、破断面に沿って四面体を分割することでより多くの破片が生成され、シミュレーションの結果が大きく異なることが分かった。例えば図 3.6 では球体が慣性に従って壁を通過していることに対して、図 3.5 では球体が衝突した要素の弾性力によって押し返されている。こういった違いが起きる理由としては、Parker 手法では各要素の大きさは変わらず分離するのみであることに対して、本手法では衝突した領域の要素が分割されて体積が小さくなり、個々の要素の弾性力が弱くなり応力がかかりやすくなるため、結果として球体に影響を与え辛く、破壊が発生しやすくなることが考えられる。

次に、計算効率についての検証結果からは要素分割によって 1 フレームの更新時間に大きな差が生まれることが分かった。またこの検証で使用した初期状態の要素数が 50 であるモデルの場合、リアルタイムで実行することは難しいことが分かった。そのため実際に使用する場合、要素数を減らしたモデルを使用するか、処理の並列化などの最適化を行う必要がある。幸い、本手法では各要素においてある程度まで破断面に忠実な亀裂を形成することができるため、モデル内に含まれる要素の密度が比較的低い場合でも現実的な破片を生成することは可能である。

総評としては、Parker 手法と比較して、より緻密な表現が可能となるが、要素の増加により

計算処理に大きな負担がかかる。そのためリアルタイムで実行することが前提となるアプリケーションに使用する場合、アルゴリズムの最適化を行った上で Parker 手法を使用する場合より要素密度が低いモデルを使用することが現実的な運用方法だと考える。

第 4 章

まとめ

ビデオゲームなどのインタラクティブなコンテンツにおいて、破壊表現は視覚的なリアリティと動的に反応を返す計算速度の両方が求められる。本研究では、リアルタイムで実行でき、現実的な亀裂や破片を生成する破壊シミュレーションの作成を目標として、破断面に忠実な亀裂や破片を生成する O'Brien 手法と、ビデオゲームで使用することを目的とした Parker 手法の中間的な手法を提案した。

本手法では、四面体有限要素法によって破壊対象の物体内部にかかる応力を計算し、その主応力によって破壊を判定する。破壊を行う場合 O'Brien 手法を基にモデルの各四面体要素を破断面に沿って分割する。この時、各要素に分割可能回数を設けることで要素が無制限に増加することを防いでいる。

検証では主に Parker 手法とリアリティと計算効率の 2 つの観点から比較を行い、結果としては要素分割によってより多くの破片と亀裂を生成し、より現実的な破壊表現を実現することができたものの、計算効率は Parker 手法と比較して大きな負担がかかることが分かった。これは要素が増加することが原因となっているため、各要素における衝突処理などの計算を並列化するなどの最適化が必要であることが分かった。

また、Parker 手法は亀裂の進行方向を既存の要素の境界に依存するという欠点と、生成された破片が四面体単体である場合に人工的な見た目になってしまう問題を各要素にグラフィックオブジェクトを割り当てることで補っている。今後の展望として、本手法でもこの方法を行うことで視覚的なリアリティを向上させることが期待できる。また、計算効率についての検証ではリメッシュ処理の違いのみで比較を行ったが、各アルゴリズムの最適化を行った上で比較することはできなかつたため、最適化した上で比較し、本手法がどこまでの要素数で実用できるか検証したい。

謝辞

本研究を進めるにあたって、最も難航したと言えるのは明確なテーマを決めることでした。破壊表現についての研究をしたいと何となく考えてはいたのですが、いざテーマを決める際には自分の知識量からこの分野に対する現状の把握ができず取り組むべき問題を決めかねていました。そんな時に渡辺先生には的確なアドバイスを頂きました。例年の生徒はどのようにテーマを決めて取り組んでいたのか、自分の場合はどのような方針を取ればいいのか、親身になって相談していただき、自分がこの研究期間にできることを理解し、明確なテーマを決めることができました。また実装の方針についての相談や、この論文の添削についてもわかりやすく指導して頂きました。ありがとうございます。

また、阿部先生にはテーマ案を模索していた時に具体的なアドバイスを頂き、この分野の場合こういったやり方になるなど、わかりやすく指導して頂きました。個別での相談をしていただく機会にはあまり恵まれませんでした。研究の新規性や実装についての相談をして頂き、自分の様々な質問に初学者でもわかりやすく対応して頂きました。おかげで実装においてその時にやるべきことを明確にすることができました。ありがとうございます。

また、同じ研究室のメンバーや高校時代からの友人たちに感謝いたします。研究室の友人たちには質問ばかりして迷惑をかけましたが、研究以外でもお互いに影響し合い、行動してきたからこそ今の自分があると思っています。ありがとう。高校時代からの友人たちには、会えずとも遠隔から通話しながら行列などの数学について教えていただき、同じ大学ではないのに相談に乗り手助けしてくれたからこそ、本研究を進めるために必要な知識を学ぶことができ、研究をスムーズに進めることができました。本当に感謝しています。

そして、ここまで学生生活を支えてくれた両親に感謝いたします。自分の将来への希望に文句一つ言わず賛成し、兄弟で私立大学に入学することを快諾してくれたからこそ自分の希望を叶えることができました。本当にありがとう。

皆様、本当にありがとうございました。

参考文献

- [1] Embark Studios. The finals. <https://www.reachthefinals.com/>. 参照: 2023.12.20.
- [2] RachelThomas and WenshuZhang. Real-time fracturing in video games. *Multimedia Tools and Applications*, Vol. 82, No. 4, pp. 4709–4734, 2023.
- [3] Epic Games. Unreal engine. <https://www.unrealengine.com/ja/>. 参照: 2023.12.20.
- [4] SatyRaghavachary. Fracture generation on polygonal meshes using voronoi polygons. In *SIGGRAPH '02: ACM SIGGRAPH 2002 conference abstracts and applications*, p. 187, 2002.
- [5] Remedy Entertainment. Control. <https://controlgame.com/>. 参照: 2023.12.20.
- [6] DICE. Battlefield3. <https://www.ea.com/games/battlefield/battlefield-3>. 参照: 2023.12.20.
- [7] SilviaSellan, JackLuong, JeticiaMattosDaSilva, AravindRamakrishnan, YuchuanYang, and AlecJacobson. Breaking good: Fracture modes for realtime destruction. *ACM Transactions on Graphics*, Vol. 42, No. 10, pp. 1–12, 2023.
- [8] MatthiasMüller, NuttapongChentanez, and Tae-YongKim. Real time dynamic fracture with volumetric approximate convex decompositions. *ACM Transactions on Graphics*, Vol. 32, No. 115, pp. 1–10, 2013.
- [9] Ubisoft. Tom clancy’s rainbow six: Siege. <https://www.ubisoft.com/en-gb/game/rainbow-six/siege>. 参照: 2023.12.20.
- [10] Julien L’Heureux. The art of destruction in rainbow six: Siege. <https://gdcvault.com/play/1023003/The-Art-of-Destruction-in>, 2016. 参照: 2023.12.21.
- [11] EricG.Parker and JamesF.O’Brien. Real-time deformation and fracture in a game environment. In *SCA '09:Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 165–175, 2009.
- [12] MatthiasMüller, LeonardMcMillan, JulieDorsey, and RovertJagnow. Real-time simula-

- tion of deformation and fracture of stiff materials. In *In Proceedings of the Eurographic workshop on Computer animation and simulation*, pp. 113–124, 2001.
- [13] 大嶋貴史, 松村誠明, 村上恭子, 鶴野玲治. Cgによる弾性体のリアルタイム剥離表現. 画像電子学会誌, Vol. 34, No. 4, pp. 327–335, 2005.
- [14] LucasArts. Star wars: The force unleashed. <https://www.starwars.com/games-apps/star-wars-the-force-unleashed>. 参照: 2023.12.20.
- [15] JamesF.O'Brien and JessicaK.Hodgins. Graphical modeling and animation of brittle fracture. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 137–146, 1999.
- [16] 物理の学校 応力テンソルの導入. <https://physics-school.com/stress-tensor/>. 参照: 2023.12.21.
- [17] MatthiasMüller. Interactive virtual materials. In *GI '04: Proceedings of Graphics Interface 2004*, pp. 239–246, 2004.
- [18] DanKoschier, SebastianLipponer, and JanBender. Adaptive tetrahedral meshes for brittle fracture simulation. In *SCA '14: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 57–66, 2014.
- [19] Bullet physics. <http://bulletphysics.org/wordpress/>. 参照: 2023.12.18.
- [20] Fine kernel toolkit system. <https://gamescience.jp/FK/>. 参照: 2023.12.18.