

3DCGにおけるカメラ視点を考慮した リアルタイム剛体アニメーションに関する研究

成田 晃^{1,a)} 柿本 正憲² 渡辺 大地² 三上 浩司² 竹内 亮太²

概要：映画やゲームの制作で用いる3次元コンピュータグラフィックスにおいて、物理シミュレーションはリアリティのあるアニメーションを生成するための重要な手段である。一般に、物理シミュレーション結果にユーザーの意図を反映させるには多くのパラメータ調整が必要で、現実には意図の反映は困難である場合が多い。本研究では、剛体アニメーションを対象とし、爆破シミュレーション等で多くの剛体破片が飛び散る場合の動きに意図を反映させる課題に取り組む。本稿では、カメラに破片が降り注ぐ場合の好ましくない動きの制御について述べる。カメラの周囲に力場を生成することにより、破片ができるだけ自然にカメラを避けるようにする剛体物理アニメーション制御手法を提案する。

キーワード：剛体シミュレーション, ユーザー制御, 爆発

On rigid body animation taking into account the 3D computer graphics camera viewpoint

Abstract: In using computer graphics for making games or motion pictures, physics simulation is an important tool to generate realistic animation. In general, it is necessary to fine-tune a number of physical parameters in order to reflect the user intention on the results of physics simulation. Thus the reflection of user intention is practically very difficult. We focus on the rigid body simulation and address the problem to reflect an intention on the movements of spattering rigid fragments broken by a simulated explosion. This paper describes how to control the undesired movements of fragments flying toward the camera. We propose a rigid body animation control method to get the fragments to avoid the camera as unobserved as possible.

Keywords: rigid body simulation, user control, explosion

1. はじめに

映画やゲームなどのための3次元コンピュータグラフィックス(3DCG)分野において、物理シミュレーションは、リアルなアニメーションを生成するためにはきわめて重要な手法の一つである。物理シミュレーションによって、剛体や流体などの挙動を計算しリアリティのある動きを容易に表現することができる。

しかしながら、一般に物理シミュレーションは、ユーザーが意図した動きを得るために多くのパラメータの調整

が必要になり、扱いづらい。本研究で対象とする物体爆破シミュレーション時の破片飛散現象表現においても、出力結果が望ましくないことがある。例えば、カメラの前で重要なアクションがあるが、カメラに対して多くの破片が降り注いでしまい、破片しか映らないシーンができてしまうといったことがある。爆発時飛散する破片はシミュレーション開始時に行先を予測することはできず、前記のような事態を回避するよう一つ一つの破片を考慮してパラメータを設定するのは困難である。

キーフレームアニメーションなどの手付けアニメーションを用いて、爆発後飛散する破片のアニメーションを生成する方法があるが、この方法は時間的コストがかかるため現実的ではない。

物理シミュレーションによりユーザーが望んだ結果を得

¹ 東京工科大学大学院 バイオ・情報メディア研究科
Graduate School, Tokyo University of Technology

² 東京工科大学 メディア学部
Tokyo University of Technology

^{a)} g311102589@st.teu.ac.jp

るためには、多くのパラメータ調整に試行錯誤が必要である。それらのパラメータの調整を行い、もっとも望ましい剛体アニメーションを得るための研究は多数行われている。

Twigg ら [3] は、さまざまな物理パラメータを変えて複数回のシミュレーションを行い、出力結果の軌跡からユーザーが目的とする結果を取捨選択する手法を提案した。Popović ら [2] は、ユーザーが物体の挙動を対話的に変更することができる手法を提案した。ユーザーが指定した挙動に合うような初期パラメータを再計算し、ユーザーは画面に表示される結果を直接変更することで目的とする出力結果を得ることができる。近藤ら [7] や 三谷ら [8] は、対話的に流体、弾性体のアニメーション制御を行った。Chenney ら [1] は、拘束条件を満たす複数のアニメーションを生成し、事前に定義したもっともらしさの評価基準により最良の結果を選ぶ手法を提案した。

しかしながら、これらの手法では繰り返し処理や結果の再計算処理が必要である。ゲームではそのような手順を踏むことは許容できない。

Twigg ら [4] は、多数の剛体モデルが望ましい配置になるように時間ステップを逆にする手法を提案した。また、川田ら [6] は、爆発後の煙粒子が望みの形状になるようステップごとに制御しつつ物理特性も保つ手法を提案した。これらの手法は、複数候補を生成することなく望ましい結果に導くことができる利点がある。しかしながら、ユーザーが、望ましい状況を固定的な形として直接指定する必要があり、状況に応じた多数の剛体モデルの制御に応用することは難しい。

今給黎ら [5] は、破壊シミュレーション後の形状を制御する手法を提案した。この手法でもユーザーは事前にばねモデルの強弱を指定する必要があり、われわれの目標とするカメラ位置に対応した破片の動きにこの手法を応用することは困難である。

このように、カメラの動きや物体の衝突状況などが予測できない変化を示すゲームシステムでは、多くの剛体破片を望ましい動きに導くことは難しい。ユーザーがどのような操作を行うかは分からないため、動的であるカメラの位置や爆発時飛散する破片の動きは一意的に決まらず、予測によるパラメータの制御は極めて困難である。無理に動きを制御すると、破片の衝突によって不自然な動きが目立ってしまう結果となる。

本研究ではこれらの問題を解決するために、動的なカメラ操作に対応し、違和感なくカメラを避ける破片のリアルタイム剛体シミュレーション手法を提案する。

2. 本研究の目標とする破片の軌跡

本研究ではリアルタイムかつ動的なカメラの動きや爆発時飛散する破片の動きのため予測できない。図1では、本研究で対象とする物体爆破シミュレーション時の破片飛散

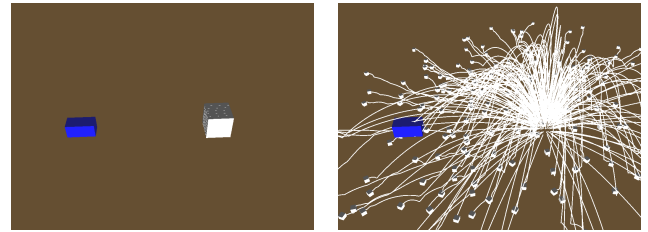


図1 爆発時飛散する破片がカメラ（青い箱）に当たってしまう例
Fig. 1 A case that some exploded fragments hit the camera (blue box)

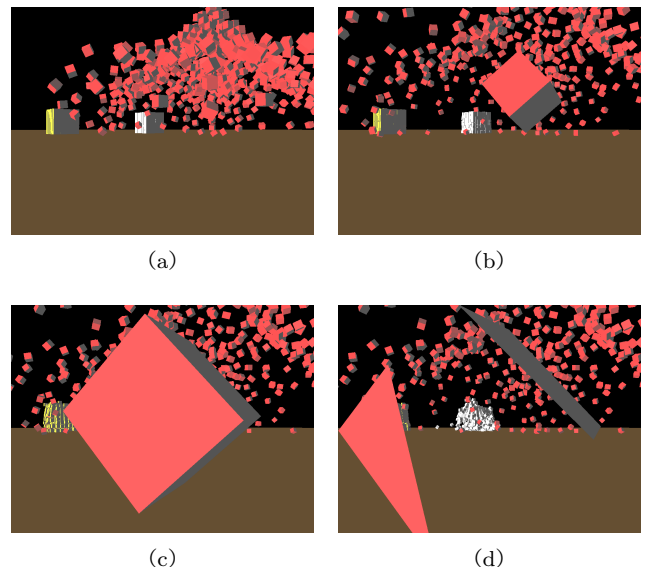


図2 破片がニアクリップ面にあたってしまう、不自然な描画になっている例
Fig. 2 A case that a fragment hits the near clipping plane, causing an unnatural rendering result

現象表現において、出力結果が望ましくないときの破片の軌跡を示す。青い直方体がカメラを示し、白い立方体が飛散する破片を示し、白い実線が試算した破片の軌跡を示す。

図1のとき、カメラと飛散した破片が衝突しているため出力結果が望ましくない。図2は、(a)、(b)、(c)、(d)と時間が進んでいるものである。図2(d)においてニアクリップ面と立方体の破片モデルが干渉しているため、立方体の一部が切り取られた不自然な描画結果になっている。

図3は、破片がカメラやニアクリップ面に衝突しないように、カメラの前に透明な壁を設置したものである。図3は、(a)、(b)、(c)、(d)と時間が進んでいる。図3(d)ではカメラの前に破片が集まってしまっても不自然な形にならないう。また透明な壁と飛散した破片が当たった場合、カメラと破片があったかのような跳ね返りが起こってしまい、望ましくない。

本研究では、映画やアニメなどでカメラに破片が飛んでくる場合、画面の上下左右方向に抜けていくという点に着目した。図4は、奥から手前に向かって破片が、実線のような軌道をとっている様子を示している。この図は映画などで画面の上下左右方向に抜けていく様子を模式的に示し

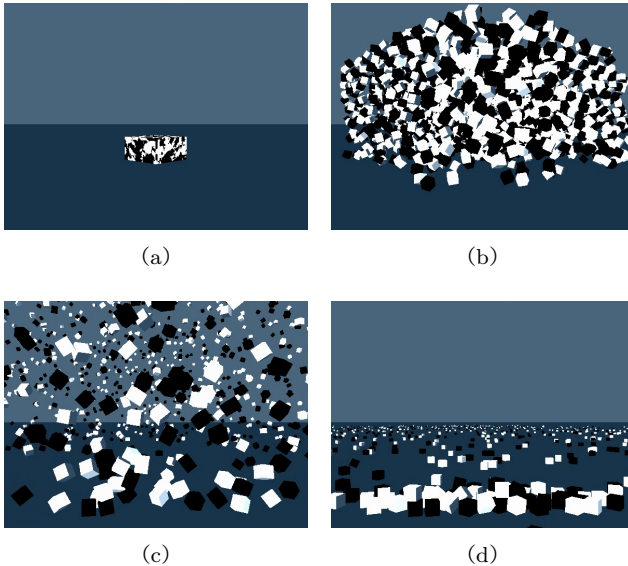


図 3 カメラの前に透明な壁を設置した場合

Fig. 3 A case that a transparent wall is placed in front of the camera

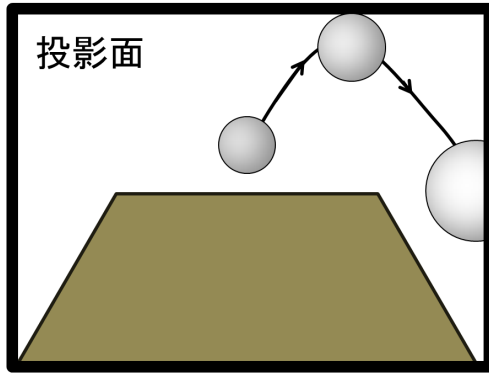


図 4 破片が画面の上下左右方向に抜けていく様子

Fig. 4 A trajectory of a fragment which goes around the frame

たものである。

本手法はゲームなどのインタラクティブなコンテンツで用いることを想定しているため、動的なカメラや、事前に予測できないタイミングで爆発飛散する破片の動きに対応する必要がある。

本研究で目標とする爆発時飛散する破片の軌跡は、次の 3 点を満たすものとする。

- (1) 動的なカメラの動きに対応している
- (2) 破片の軌道を変える際、破片の動きの違和感をなくするため、カメラの上下左右方向に破片を逸らす
- (3) カメラに対して破片が飛んでくるとき、物理的に多少不自然でもカメラに破片が当たらないことを優先する

本研究では、カメラに降り注ぐ破片に対して力を加えることで破片の動きを制御し、カメラから違和感なく逸らすような破片の軌跡を描くリアルタイム剛体シミュレーション手法を提案する。剛体が、動くカメラに当たらずに、カメラの上下左右方向に逸れるような軌道をとることを目標

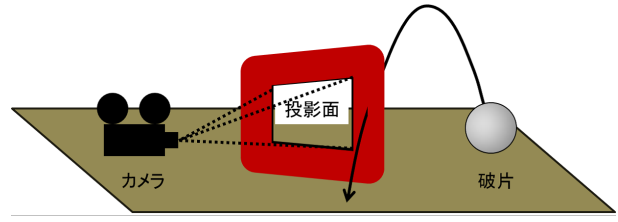


図 5 本研究で理想とする破片の軌道

Fig. 5 An ideal fragment trajectory in the proposed method

とする。図 5 は、爆破直後はカメラに向かう破片が、最終的にはカメラの上下左右方向に逸れるような軌道をとる様子を模式的に示したものである。

日常にある物体のほとんどは反発係数が極端に高いものは少ないため、多くの場合、一度着地した破片が大きく跳ね上がることはほとんどないを考える。そのため本手法では、カメラに影響はないとみなし、一度地面に着いた破片は動き制御の対象としない。

3. 提案手法

3.1 衝突予想計算

爆破直後の破片が地面に落ちるまでの時間にカメラが移動可能な領域を考え、その領域と破片の予想軌跡との交差判定を行うことで、カメラに当たる可能性がある破片を検出する。提案法ではカメラの移動最大速度 V_{cam} は決まっているものとする。爆破時の時刻を $t = 0$ とする。時刻 t における破片 p の位置を $\mathbf{P}_p(t) \equiv (x_p(t), y_p(t))$ 、速度を $\mathbf{V}_p(t) \equiv (V_p^x(t), V_p^y(t))$ とすると、破片 p が描く放物線の軌跡は次の式によって表される。

$$\begin{aligned} x_p(t) &= x_p(0) + V_p^x(0)t \\ y_p(t) &= y_p(0) + V_p^y(0)t - \frac{1}{2}gt^2. \end{aligned} \quad (1)$$

ただし、 g は重力加速度の値である。このとき、破片 p が地面に落ちる予想時刻 t_p^G は、式 (1) で $y = 0$ とおいて t についての二次方程式を解くことで求まる。このとき、予想時刻までにカメラが動く最大距離 $R(t)$ は

$$R(t) = V_{cam}(t_p^G - t) \quad (2)$$

となる。時刻 t におけるカメラ位置を $\mathbf{P}_{cam}(t) \equiv (x_{cam}(t), y_{cam}(t))$ とすると、爆破時点 ($t = 0$) から時刻 t までにカメラが動きうる領域は次の式で定義できる球となる。

$$|\mathbf{P}_{cam}(t) - \mathbf{P}_{cam}(0)|^2 = R(0)^2. \quad (3)$$

式 (1) と式 (3) を連立させると、 t に関する 4 次方程式となる。これを解いて t の範囲が $[0, t_p^G]$ となる解が存在すれば、破片 p はカメラと衝突する可能性があるとして $t = 0$ の時点で推定できる。この時の t の解 t_p^C を最短衝突予想時刻と呼ぶことにする。この一連の計算を $t = 0$ における衝突予

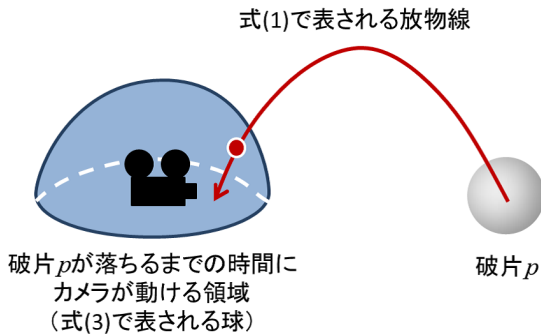


図 6 衝突予想計算

Fig. 6 Collision prospect computation

想計算と呼ぶことにする。また、このようにしてカメラと衝突する可能性がある判定された破片を**衝突候補**と呼ぶことにする。

図 6 は、衝突予想計算により衝突の可能性が予想され、最短衝突予想時刻が求められる様子を示す。このときの破片 p は、次のタイムステップでも衝突候補となる。

爆破直後の衝突候補を求めるには、飛散するすべての破片について $t = 0$ における衝突予想計算を行う。理論的には、爆破後に放物線を描く破片の初速度は、爆破に伴う力の作用が終わった時点での速度とすべきであるが、提案法で利用する物理シミュレーションでの計算では、 $t = 0$ の瞬間で爆破力の作用が終わり、放物運動の初速度が求められる。そのため、式 (1) における $\mathbf{V}_p(0)$ は物理シミュレーションにより得られる値を使う。

$t = 0$ における衝突予想計算で衝突候補とならなかった破片については、以降 ($t > 0$) の各タイムステップでの衝突予想計算は行わない。タイムステップ n での時刻 $t = t_n > 0$ における衝突予想計算は $t = 0$ のものと同様である。破片の放物線を示す式 (1) についても、カメラが動く範囲を示す式 (3) についても、 $t = 0$ を $t = t_n$ に置き換えた式を用いることとなる。各タイムステップでの衝突予想計算の繰り返しによって、衝突候補は少しずつ減ってゆく。

図 7 は、タイムステップが進むにつれて衝突候補が減ってゆく様子を示す実験結果のグラフである。

3.2 衝突候補に対する力の印加

飛散した破片のうち、衝突候補に対しては意図的に力を加え、図 5 で赤く示した領域、すなわち投影面の枠の外側に飛んでいくように制御することが望ましい。このとき加える力が大きすぎると不自然な動きになるため、最小限の力で画面外へ衝突候補を逸らす必要がある。

基本的な考えは、最短衝突予想時刻まで余裕があれば小さな力を、逆に最短衝突予想時刻が迫っていれば大きな力

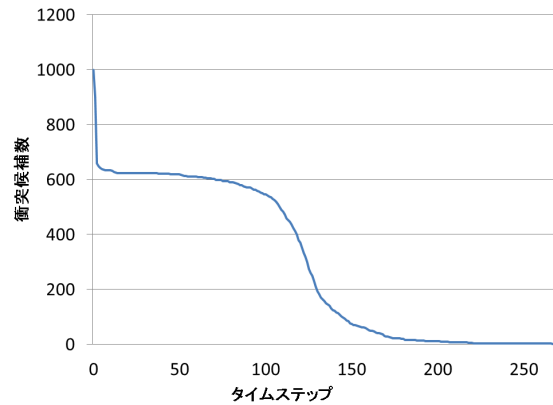


図 7 タイムステップと衝突候補数との関係

Fig. 7 The relationship between time steps and collision candidates

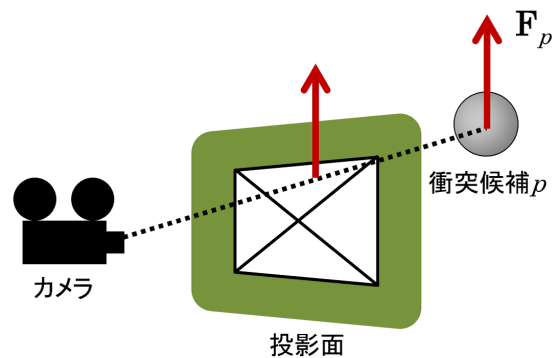


図 8 衝突候補に加える力の向きの決定

Fig. 8 The direction of the force to add to a collision candidate

を加える、というものである。加える力の向きは、衝突候補の投影面上での位置を検出し、投影面の四辺 (上下左右) のうちもっとも近い辺に向かう垂線の向きとする。たとえば、図 8 に示すように上辺がもっとも近ければ世界座標での対応する向き (上向き) の力を加える。

衝突候補 p に加える力を $\mathbf{F}_p$ 、 p の投影面位置からもっとも近い辺への垂線に対応する世界座標での向き (単位ベクトル) を $\mathbf{D}_p$ 、 p の最短衝突予想時刻を t_p^C とすると、

$$\mathbf{F}_p = \frac{k}{t_p^C - t} \mathbf{D}_p \quad (4)$$

が成り立つ。ここで k はユーザー設定の比例定数、 t は現在時刻である。

4. 結果

提案手法をプログラムによって実装し、物理シミュレーションを行い、本手法の有用性を検証した。実装には PhysX を用いた。

本シミュレーション実験では、カメラの正面にある立方

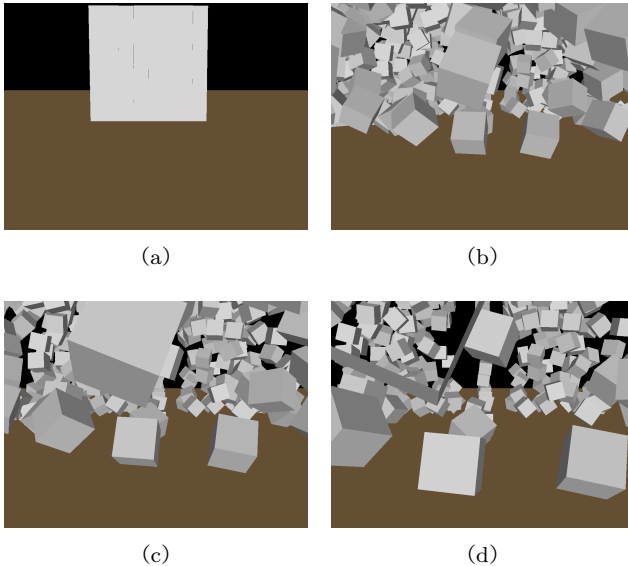


図 9 本手法を使わない場合の破片の飛散する様子

Fig. 9 Fragment spattering without our method

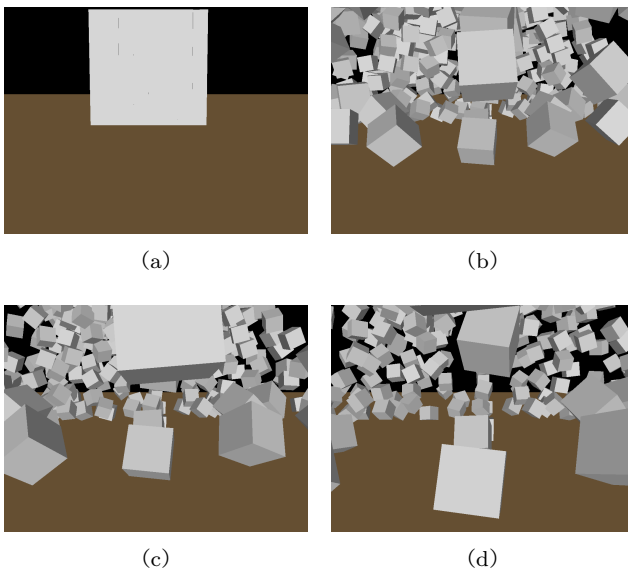


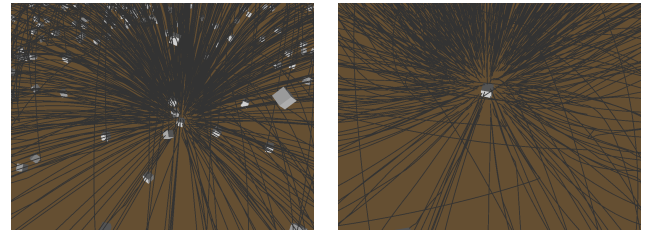
図 10 本手法を使った場合の破片の飛散する様子

Fig. 10 Fragment spattering with our method

体が爆発して更に小さい立方体に分かれて飛散する。カメラは正面に向かって移動させている。立方体は最大で 512 個の立方体に分かれるものとしている。

図 9 は、提案手法を用いず、爆発時の飛散する破片シミュレーションを行った場合の実行結果である。カメラと飛散した破片が当たってしまい、不自然な結果になっている。図 9 (d) では、ニアクリップ面に当たっているため画面左上の破片に大きな穴があいたような望ましくない描画結果になっているのがわかる。

図 10 は、提案手法を用いて、爆発時の飛散する破片シミュレーションを行った場合の実行結果である。飛散した破片はカメラに当たらずに画面の上下左右方向に逸れている。



(a) 本手法を用いない場合の軌跡 (b) 本手法を用いた場合の軌跡

図 11 カメラ位置から見た飛び散る破片の軌跡

Fig. 11 Trajectories of fragments viewed from the camera position

図 9, 図 10 は、いずれも (a), (b), (c), (d) と時間が進んでいるものである。

図 11 は、提案手法を用いて、爆発時の飛散する破片シミュレーションを行った場合の結果を上から見た様子である。図中に描かれている線は、飛散した破片の軌跡である。

図 11 (a) では、破片には何も力が加わっていないためカメラに向かってくる軌跡が密に描かれている。図 11 (b) では、提案手法により破片に力が加わったため、カメラに直接向かってくる軌跡が減っていることがわかる。これに対応する図 10 で示したように、カメラから見た描画では違和感のないものになっている。

5. まとめと今後の課題

本研究では、動的なカメラに向かって飛んでくる破片に対して単純な力を加えるのではなく、逸らす位置も重要と考え、逸らす位置を考慮したリアルタイム剛体シミュレーション手法を提案した。カメラの動きに対応して破片の軌道を変える際、動きの違和感をなくすため、上下左右のうち最も適切な方向に破片を逸らす。さらに、衝突の危険性が高いほど強い力で破片の軌道変更制御を行うようにした。これにより、物理的に多少不自然でもカメラに破片が当たらないことを優先するという動きが実現できた。

今後の展望として、破片がカメラから逸れて行くだけでなく、破片が飛んでくるまでの”タメ”の表現ができないかを検討する。また、破片がカメラの上下左右のうちどの辺に多く現れるかの割合や飛んでくる頻度を、ユーザーの意図に応じて調整できるようなものにしたいと考えている。

参考文献

- [1] Cheney, S. and Forsyth, D. A.: Sampling Plausible Solutions to Multi-body Constraint Problems, Proc. SIGGRAPH 2000, pp. 219-228 (2000)
- [2] Popović, J., Seitz, S.M., Erdmann, M., Popovic, Z. and Witkin, A.: Interactive manipulation of rigid body simulations, ACM Transactions on Graphics, Vol. 26, No. 3 (Proc. SIGGRAPH 2000), pp.209-217 (2000)
- [3] Twigg, C.D. and James, D.L.: Many-worlds browsing for control of multibody dynamics, ACM Transactions on Graphics, Vol. 26, No. 3 (Proc. SIGGRAPH 2007),

Article 14(2007)

- [4] Twigg, C.D. and James, D.L.: Backward Steps in Rigid Body Simulation, AMC Transactions on Graphics, Vol. 27, No. 3 (Proc. SIGGRAPH 2008), Article 25 (2008)
- [5] 今給黎隆, ヘンリー・ジョハン, 西田友是:破壊後の形状の制御を伴う破壊シミュレーション, 画像電子学会誌, Vol. 38, No. 6, pp. 449-458 (2009)
- [6] 川田玄一, 金井崇:物理モデルを考慮した爆発のモデリング, Visual Computing /グラフィクスと CAD 合同シンポジウム, No.20 (2011)
- [7] 近藤 亮, 金井 崇, 安生 健一:物理指向補間による弾性体アニメーションの制御手法, Visual Computing /グラフィクスと CAD 合同シンポジウム, pp. 5-10 (2006)
- [8] 三谷 純, 五十嵐 健夫:流体シミュレーションを統合した対話的な形状設計手法, In Proc. Workshop on Interactive Systems and Software, pp. 25-28 (2008)