



Title	特殊相対論の効果を考慮したシミュレータの研究と物理学への応用
Author(s)	安田, 航一郎
Citation	令和元 (2019) 年度学部学生による自主研究奨励事業研究成果報告書. 2020
Version Type	VoR
URL	https://hdl.handle.net/11094/75977
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

2019年度大阪大学未来基金【住野勇財団】学部学生による自主研究奨励事業研究成果報告書					
ふりがな 氏名	やすだ こういちろう 安田 航一朗	学部 学科	理学部物理学 学科	学年	1年
ふりがな 共同 研究者氏名		学部 学科		学年	年
					年
					年
アドバイザー教員 氏名	おだきんや 尾田欣也	所属	大阪大学大学院 理学研究科物理学専攻		
研究課題名	特殊相対論の効果を考慮したシミュレータの研究と物理学への応用				
研究成果の概要	研究目的、研究計画、研究方法、研究経過、研究成果等について記述すること。必要に応じて用紙を追加してもよい。(先行する研究を引用する場合は、「阪大生のためのアカデミックライティング入門」に従い、盗作剽窃にならないように引用部分を明示し文末に参考文献リストをつけること。)				
<p>1, 研究の要約</p> <p>特殊相対性理論は、四次元時空での物体の運動の記述と、ローレンツ変換を施すことによって分かる相対論的効果について論じた理論である。こうした効果は光速に近くなるときに顕著に現れるものであり、人間が直接感じることは現実上困難である。このときに効果的であるのが、コンピュータシミュレーションによる効果の再現である。本研究では、尾田氏、中山氏の開発した「相対論的ゲーム」や、MIT Game Labが開発した「Open Relativity」のスクリプトと動作原理を研究し、こうした相対論的運動に限らず、他の相対論的効果を考慮に入れる必要がある電磁場による物理現象に対して有効なシミュレーションをUnityで作った。その結果、ある速度をもって運動する物体から見える電場のベクトルが変化することを可視化することに成功した。</p>					
<p>2, 研究の内容</p> <p>(1) 研究動機</p> <p>私は大阪大学理学部物理学学科主催の講義「物理学セミナー」において相対論的ゲームを作るという講義をとった。この時、相対論的ゲームの開発者である尾田欣也准教授の講義を通して相対論的運動をゲームプログラムを通して理解するというアイデアに興味を持ち、自分でも作ってみようと考えた。この講義で気づいたこととして、このプログラムでは他のオブジェクトはプレイヤー自身に影響を与えていないことがあり、私は自分が開発するプログラムでは電磁場がローレンツ変換でどう変換されるかを可視化したいと考えた。</p>					
<p>(2) 先行研究の要約</p> <p>代表的な先行研究としては2つある。一つは大阪大学理学研究科物理学専攻の尾田欣也准教授が開発した相対論的ゲームと、MIT Game Lab.が開発した Slower Speed of Light をもとに作られた開発ソフト Open Relativity である。これらでは、相対論的な運動の記述が可能な上、光のドップラー効</p>					

果も出力できることが特徴である。特に Open Relativity では花火や別の運動するボールなどを出力することで時間の流れが遅く感じられることも表現している。

(3) 研究の方法と過程

本研究では、大きく分けて以下の 3 つの段階を通して行った。まず、先行研究として作られた、「相対論的ゲーム」と Open Relativity の 2 つのプログラムのスクリプトを読み、特殊相対性理論の考え方がどういった記述のされ方がされているのかに加え、スクリプト同士の共通点と相違点を調べ、改良すべき点についても探し、必要な数学もその都度勉強した。「Unity でわかるゲーム数学」で主に回転変換の四元数を用いた表現を学ぶとともに、「相対性理論入門講義」ローレンツ変換の行列による表記法と、本テーマの中心的役割を果たす相対論的現象を学んだ。次に得られたシミュレーションの動作原理から、改めて自作の相対論効果シミュレータの開発を行った。ただし、時間成分に関しては直接ローレンツ変換するのではなく、ローレンツ変換後の速度 u を用いて、

$$t = \sqrt{1 + u^2}$$

と表せるので、ゲーム上のプレイヤーが感じる時間の流れとして用いた。

開発環境は Unity を用いて、開発言語は C# にした。ゲームの内容として、まずはこれまで同様のスケールで開発し、出来たプログラムを参考にした 2 つのプログラムと比較した。特に改良すべき点が改善されているのかを調べた。ここにおいても、「Unity でわかるゲーム数学」を用いた。最終的に出来上がったプログラムを用いて時間的に変化しないような電磁場をローレンツ変換した。ここでローレンツ変換の方法は電磁場の強さという反対称テンソルにローレンツブーストを左から作用させることで計算した。つまり、電磁ポテンシャル A^μ 、ローレンツブースト Λ^μ_ν としたとき、

$$F^{\mu\nu} = \partial^\mu A^\nu - \partial^\nu A^\mu$$

$$F'^{\mu\nu} = \Lambda^\mu_\sigma \Lambda^\nu_\lambda F^{\sigma\lambda}$$

と表し、変換は Slerp(球面線形補間)を用いて行った各物体の座標のローレンツ変換も同様に行った。では、プレイヤーから物が見えるというのはどう定義したら良いのかというと、そのためには時空間上での新たな距離を定義する必要がある。相対性理論において、それは時空間上のプレイヤーの 4 元位置ベクトルとゲームオブジェクトの 4 元位置ベクトルを用いて次のようなローレンツノルムを取ることで定義される。時刻 $t \in \mathbb{R}$ として、ローレンツ内積

$$\langle u, v \rangle = u_0 v_0 - \mathbf{u} \cdot \mathbf{v}$$

の $u=v$ の場合で定義されるローレンツノルムの 2 乗を用いて

$$(Lnorm)^2 = \langle t\vec{n} + \vec{X}_{GO} - \vec{X}_p \rangle^2$$

これは t に関する二次方程式であるので、その解はオブジェクトが光子やタキオンではないという仮定のもとでは、

2 つ存在している。ここで 2 つ出てきたうちの小さい方がプレイヤーの Past Light Cone との交点で、大きい方が Future Light Cone である。つまり、上図のような状況の 4 次元時空ではプレイヤーにとって見えるとはあるオブジェクトがその世界線上のプレイヤーの Past Light Cone と交点を持つような座標にいるということである。これは世界系での座標のことであるので、シミュレータ画面で出力する座標というのは、この位置ベクトルのローレンツ変換で与えられる。ゆえに

$$\vec{X}_p = L(-u)\vec{x}_p$$

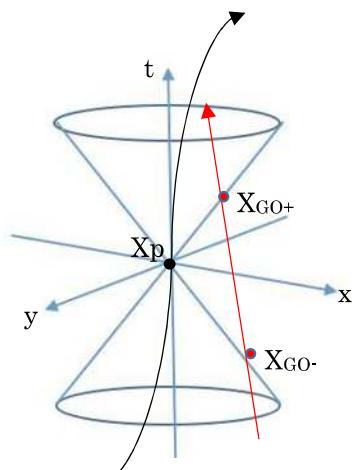


図 1:プレイヤーの光円錐とオブジェクトの世界線

$$\overrightarrow{x_{GO}} = \mathbf{L}(u) \cdot \overrightarrow{X_{GO}}$$

で変換された \vec{x} をシミュレータ上での新たなオブジェクトの座標として出力する。

詳細なスクリプトを添付してあるのでぜひ色々触ってみてほしい。

(4) 研究結果と今後の展望

本研究の結果新たに電磁場ベクトルがある速度をもって運動する物体にとってどう見えるのかということをプログラムによって出力できるようになった。これはそもそも目に見えない電磁場がローレンツ変換の影響を受けて変化する様子を矢印の向きによって出力できたものである。ただ、この結果は時間によって変化しない電磁場という仮定のもとで出てきたものであり、実際の電磁場は相互作用を受けながら、時間的に変化するものである。そのため、ローレンツ変換前の電磁場もローレンツ変換後の電磁場も本研究での結果と異なることが予想される。今後の研究においてはまず電磁場ポテンシャル $A^{\mu \nu}$ の各成分に時間変化の成分を加えることで電磁場ベクトル \mathbf{F} を変化させてプログラムに組み込むことが必要である。今後の展望としては、光に対して粒子が高速度で運動するような現象例えば超重量恒星周辺の星間ガスや粒子加速器内部での運動物体周辺の電磁場といったもののシミュレーションの開発、加えて他の電気量を持った粒子との相互作用を取り入れたプログラムの開発も行いたいと考える。

(5) 参考文献・web リンク

風間洋一 相対性理論入門講義 培風館 2017 P58-73, P125-138

高原文郎 特殊相対論 培風館 2012 P101-109

岡真 電磁場の古典論 培風館 2009 P93-108

Unity でわかるゲーム数学 P2-108, P300-363

<https://docs.unity3d.com/ja/current/Manual/QuaternionAndEulerRotationsInUnity.html>

<https://qiita.com/Teach/items/8d1cdb90a3b533707c9a>

https://qiita.com/tsukasa_wear_parker/items/09d4bcc5af3556b9bb3a

<http://htsuda.net/archives/1702>

<http://unity-indie-dou.hatenablog.com/entry/2017/08/19/100000>

https://tech.pjin.jp/blog/2016/02/16/unity_vector3_1/

<https://qiita.com/broken55/items/33ac4ab55744b2e5e562>

<https://wiki.unity3d.com/index.php/DrawArrow>

<https://answers.unity.com/questions/1100566/making-a-arrow-instead-of-linerenderer.html>