

OffscreenCanvas を用いたシミュレータ教材の軽量化

1532040 岡本 悠祐 指導教員 須田 宇宙 准教授

1 背景

シミュレータ教材は不可視現象を可視化する教材である。そのため、イメージが困難な事象への理解促進を促す手法として有用である。e-Learning の普及に伴いシミュレータ教材の活躍の場が増加し、様々な分野に対応したシミュレータ教材と効率的な開発手法が必要とされている。

磁場や音場などの波形を可視化するシミュレータにおいて FTDT(Finite-difference time-domain) 法が広く用いられているが演算回数が多いことから処理速度は決して速くない。

本研究室では昨年、FTDT 法を用いたシミュレータ教材に GPU を適用し CPU との処理速度を比較する研究を行った。その結果、大幅な処理速度向上が得られたが、ソースコードの変更箇所が多く、複雑なため容易に実装できないという問題点が浮上した。

その問題点は JavaScript で Worker を活用し、描画を行う OffscreenCanvas を利用することで緩和され则认为る。

本研究では OffscreenCanvas を利用した FTDT 法を用いたシミュレータ教材の制作を行い、ソフトウェアの生産性と処理速度を昨年制作したシミュレータ教材と比較することで、有用性の検証を行うことを目的とする。

2 制作したシミュレータ教材

本研究では OffscreenCanvas を利用したシミュレータ教材の有用性の検証を行う。プログラムには昨年本学卒業論文である「GPU 利用によるシミュレータ教材の演算速度」を拡張し、処理速度とソースコードの比較を行う。

図 1 は本研究で制作した複数の音源から発生する音場を可視化したシミュレータである。本研究では図 2 のようにキャンバスを左右に分けることにより、負荷を分散させた。

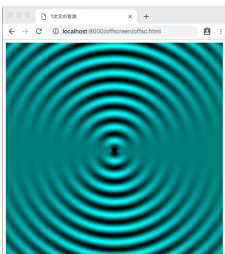


図 1: 制作したシミュレータ

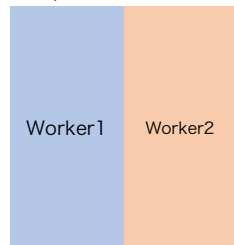


図 2: キャンバス分割イメージ

3 検証方法

昨年のシミュレータと演算速度を比較、変更箇所を比較することで OffscreenCanvas の実用性の検証を行う。

1,000 回計算する時間を計測、10,000 回まで行った。比較

に昨年のプログラム 2 つと OffscreenCanvas, Worker を利用した 4 つを用いた。

計測に使用した PC のスペックを表 1 に示す。

表 1: 使用した Mac のスペック

OS	macOS High Sierra
メモリ	32GB
CPU	3.2GHz Intel Core i5
GPU	NVIDIA GeForce GT 775M 1024MB

4 検証結果

昨年制作されたシミュレータ教材では CPU と GPU の処理速度の比較を行った。ここでは昨年に制作され、CPU で動作させた方を CPU、GPU で動作させた方を GPU と呼ぶ。CPU の処理速度を基準とし、各処理速度を式 1 に当てはめて求めた。

キャンバスを分割した WebWorker と OffscreenCanvas は処理速度が遅い方を参照する。その結果を表 2 に記した。

$$\text{処理効率} = \frac{\text{各手法 10,000 回の処理速度 [ms]}}{\text{基準 10,000 回の処理速度 [ms]}} \quad (1)$$

表 2: 各種法の処理速度比較

GPU	1: 0.810988
OffscreenCanvas	1: 0.80789763
WebWorker	1: 0.795556697

OffscreenCanvas の変更箇所は分割したキャンバスの範囲指定と、キャンバスの生成元である HTML のキャンバスを増やすだけで、ソースコードの変更点はほとんどなかった。

Worker のみと違い worker を宣言するメインではデータのデータの引き渡しがないだけでなく、左右のキャンバスを整合性をとる必要がないことから OffscreenCanvas は Worker より実装が容易である。

5 終わりに

本研究では OffscreenCanvas の FTDT 法を用いたシミュレータへの有用性を検証した。OffscreenCanvas はまだ出たばかりの技術のため、最適化されていない可能性があり、今後のさらなる改良と、知識の共有が必要である。

参考文献

- [1] 土井 淳: “ソフトウェアの生産性に着目した GPU プログラミングの方法論に関する研究動向”, 電子情報通信学会誌, Vol.101, pp1219-1223(2018)