

GPUを用いた光子輸送シミュレーションの高速化とそのマンモグラフィ画像への応用

SEZAI, Shota / 瀬在, 翔太

(出版者 / Publisher)

法政大学大学院理工学研究科

(雑誌名 / Journal or Publication Title)

法政大学大学院紀要. 理工学研究科編

(巻 / Volume)

63

(開始ページ / Start Page)

1

(終了ページ / End Page)

8

(発行年 / Year)

2022-03-24

(URL)

<https://doi.org/10.15002/00025379>

GPUを用いた光子輸送シミュレーションの 高速化とそのマンモグラフィ画像への応用

ACCELERATION OF A PHOTON TRANSPORTATION WITH A GPU AND ITS APPLICATION TO MAMMOGRAPHY IMAGES

瀬在翔太

Shota SEZAI

指導教員 尾川浩一

法政大学大学院理工学研究科応用情報工学専攻修士課程

Low-energy X-rays used in mammography cause a large effect by coherent scattering, and the resulting scattered photons decrease the quality of mammograms in terms of the contrast resolution. When we calculate these effects by using a Monte Carlo simulation with a general purpose code such as EGS, the complexity of the phenomenon increases the simulation time. To solve this problem, we developed a parallel processing code using a GPU and compared the performance of our code with these of EGS5. As a result, we succeeded in achieving 23 times faster computation with the same accuracy as EGS5. Then we applied our code to calculate the scattered photons with a numerical simulation phantom, and tried to estimate the scattered photons with a machine learning method. The results showed that we can estimate the scattered photons with a MSE of about 3.3%.

Key Words : X-ray, GPU, Monte Carlo simulation, mammograph

1. はじめに

現在、日本では乳がんによる死亡者数が増加している。これにはマンモグラフィ検診による早期発見が有効である。しかし、マンモグラフィでは低エネルギーのX線を扱うため、コヒーレント散乱が起こる割合が多く、撮影画像に散乱線が多く含まれ、画像のコントラストが低下し、小さな腫瘍や石灰化を診断で見落とす原因となる。

また、マンモグラフィ装置は、コンピュータシミュレーションにより開発される。この過程で用いられるのが、モンテカルロ法による光子輸送シミュレーションである。これは乱数により、光子の移動方向、移動距離、相互作用の種類を確率的に決定することによって媒質の中を通過する光子の挙動をシミュレーションするものである。これは実際のX線照射に近い挙動を再現できる優れた技術であるが、計算コストが高い。特に、低エネルギーのX線で多く見られる、コヒーレント散乱の計算が複雑であり、精度の高いシミュレーションには膨大な時間を要する。しかし、医学物理学分野の光子輸送シミュレーションで使用されるEGS5のような汎用ソフトにはGPU向けのプラットフォームは存在しない。

そこで本研究では、マンモグラフィ画像の散乱線推定

のために、GPUで並列処理可能な低エネルギー向け光子輸送シミュレーションコードの開発し、EGS5よりも高速な計算を行えるようなプログラムを作成した。さらにこれを用いて、マンモグラフィで発生する散乱線を定量評価し、自作コードで取得したデータセットを用いた機械学習を実施し、散乱線推定を行い有用性を評価した。

2. マンモグラフィ装置と汎用シミュレーションコード

(1) 乳がんとマンモグラフィ

現在日本では乳がんによる死亡数が年々増加している。乳がんはがんのなかでも、日本女性がかかる割合(罹患率)がトップであり、その罹患率も増加し続けている。生涯のうち乳がんになる女性の割合は、60年前は50人に1人であったが、現在は9人に1人とされており、年間9万人以上が乳がんにかかる。それに伴い、年間約1万5,000人が死亡している。

年次別の乳がんによる死亡数の推移を以下、図 1 に示す。

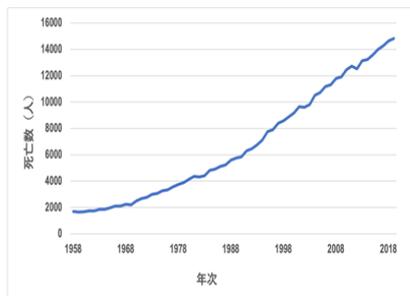
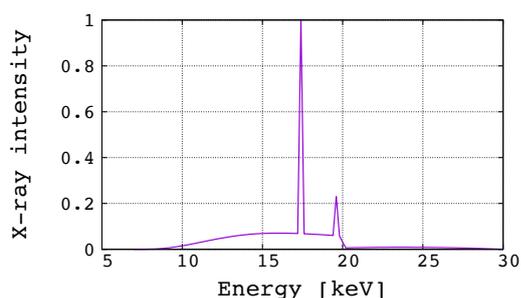


図 1 乳がんによる死亡数の推移

(2) マンモグラフィ装置について

マンモグラフィは、X 線を用いて左右の乳房を上下方向と斜め方向から撮影する検査方法であり、手で触れることのできない小さなしこりや、早期がんのサインである石灰化を映し出すことができ、がんの早期発見に対する有用性が認められている。線源の特徴としては、一般 X 線撮影では管電圧が 80 kV 程度であるのに対し、マンモグラフィでは 30 kV 程度である。X 線のエネルギーが低いほど乳房組織と病巣の線減衰係数の差が大きくなるため、コントラストの高い画像を取得するためには低エネルギーの X 線が必要となる。しかし、X 線のエネルギーが低くなりすぎると、軟部組織による X 線の吸収が大きくなってしまいうため、線量と画質を両立した管電圧 30 kV 程度のエネルギーの X 線がマンモグラフィでは利用される[1]。以下に実際のマンモグラフィで用いられる X 線のスペクトルを示す。



Mo target with a Mo filter (0.03mm)

図 2 マンモグラフィの X 線スペクトル

このようにマンモグラフィでは低エネルギーの X 線を用いるため、相互作用としてコヒーレント散乱の割合が多くなる。

以下に、水媒質で起こる相互作用の割合を示す。

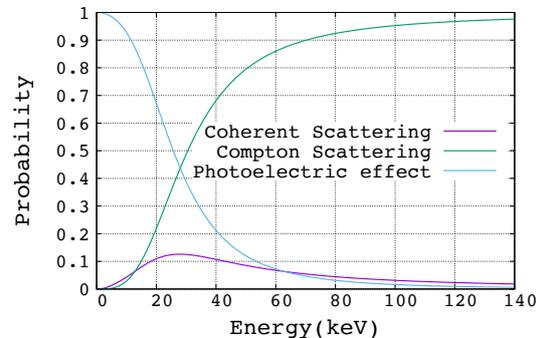


図 3 相互作用の割合

コヒーレント散乱による散乱線がコントラストの低下を招くことに加え、複雑な計算によりシミュレーション時間が膨大になるという問題がある。

(3) EGS

放射線撮影装置の設計のためにはコンピュータによるシミュレーションによる性能評価が必要である。そのシミュレーションコードとして最も使用されるのが、という汎用のソフトである Electron Gamma Shower (EGS) である[2]。この EGS には GPU 向けのコードが存在せず、コヒーレント散乱などの複雑な計算が多い低エネルギー向けのシミュレーションには膨大な時間を要する。そのため、本研究ではマンモグラフィ画像の散乱線の推定の手段とし、GPU で並列計算可能な光子輸送シミュレーションの高速化コードを作成する。そこで精度と計算時間を EGS5 と比較する。

3. GPU と CUDA

(1) GPU

GPU とは Graphics Processing Unit の略であり、多数のコアを持ち、並列処理を行うことで高速計算が可能なハードウェアである。GPU は CPU に比べコア単体の処理能力は低い、多数のコアにより、複数の単純計算を同時に行うのに適している。本研究のシミュレーション計算では、Tesla P100(Ampere)を用いる。

(2) CUDA

CUDA とは Computed Unified Device Architectures の略であり、NVIDIA 社が提供する GPU コンピューティング向けの統合開発環境である。これはプログラム記述、コンパイラ、ライブラリ、デバッガなどから構成される。次に一般的な CUDA プログラムの処理の流れについて述べる。CUDA では CPU (ホストとも呼ばれる) と GPU (デバイスとも呼ばれる) を組み合わせて使用する。またカーネル関数と呼ばれる GPU で実行される関数は並列計算処理が

可能な関数であり、CPU で呼び出し、実行される。

プログラム処理の流れを以下に示す。

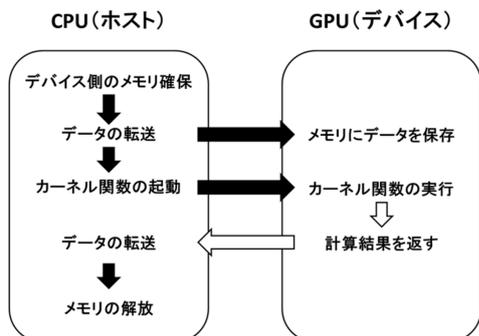


図 4 CUDA プログラム処理の流れ

上図の処理ではスレッド、ブロック、グリッドという概念を考える。スレッドとはカーネル関数を動作させたときの多数のプログラムを実行する最小単位である。各スレッドには通し番号が振られ、全てのスレッドが同一のコードを処理する。次にブロックとは複数スレッドのまとまりである。x,y,z の三次元以下に設定でき、管理される。スレッド同様、各ブロックに通し番号が振られる。最後にグリッドとは、複数ブロックのまとまりである。1つのカーネルは 1 つのグリッドで表現される。ブロック同様、x,y,z の三次元以下に設定でき、管理される。本研究では、光子の挙動計算をスレッドに割り当てて、並列化を行う。

4. 光子輸送計算

本研究のシミュレーションではプログラムにより、光子の移動距離、相互作用などの計算を行う。

(1) コヒーレント散乱

本研究においてシミュレーション高速化の要となるコヒーレント散乱の実装について説明する。電磁波が自由電子に当たると、電磁場によって物質内の自由電子は揺り動かされる。このとき、振動する電子の速さが光速に比べて十分無視できる程度であれば、振動させられた電子から同じ振動数の電磁波が放射される。このような、自由電子による電磁波の散乱をコヒーレント散乱という。これはシミュレーションでの数式が非常に複雑になり、計算に膨大な時間を費やす大きな要因となる。シミュレーションコードを作成する上で、実装したコヒーレント散乱の計算式を以下に示す。R を電子半径、k をエネルギー、 $F_T(q)$ を形状因子とすると、

$$\frac{d\sigma(\theta)}{d\Omega} = \frac{r_0^2}{2} (1 + \cos^2 \theta) [F_T(q)]^2 \quad (1)$$

$$q = 2k \sin \frac{\theta}{2} = \sqrt{2}k\sqrt{1 - \cos \theta} \quad (2)$$

ここで、 $d\Omega = 2\pi d(\cos \theta)$ 、 $\mu = \cos \theta$ 、 $q^2 = 2k^2(1 - \mu)$ を用いると、次の式 (3) が得られる。ただし、 $A(q^2)$ は式(4)で表される。

$$\begin{aligned} \frac{d\sigma(q^2)}{dq^2} &= \frac{\pi r_0^2}{k^2} \left(\frac{1 + \mu^2}{2} [F_T(q)]^2 \right) \\ &= \frac{\pi r_0^2}{k^2} A(q_{max}^2) \left(\frac{1 + \mu^2}{2} \right) \frac{[F_T(q)]^2}{A(q_{max}^2)} \end{aligned} \quad (3)$$

ここで、

$$A(q^2) = \int_0^{q^2} [F_T(q)]^2 d(q^2) \quad (4)$$

である。

また、形状因子媒質によって定められる固有の値である。以下に水とカルシウムの形状因子[3]を示す。

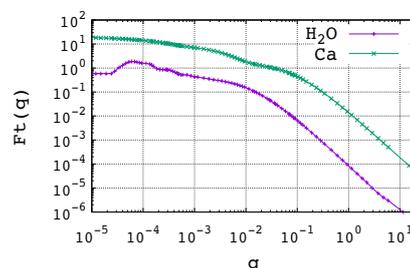


図 5 水とカルシウムの形状因子

5. シミュレーション

(1) 自作コードと EGS の比較

a) GPU 向けの自作コード

CUDA を用いて GPU 向けのシミュレーションコードを作成した。光子 1 つ 1 つの挙動計算を for 文で繰り返し行うことでシミュレーション時間が増加する原因があるところに着目し、この光子 1 つの挙動計算を 1 スレッドに割り当てて高速化を図った。今回は光子数 1 億個を想定したため、光子 1 つの処理を 1 つのスレッドに割り当てられるように、グリッドサイズ×ブロックサイズ(=総スレッド数)を 1 億個に設定した。

b) シミュレーション方法

自作コードと EGS5 の結果を比較するために、簡単なジオメトリでモンテカルロ光子輸送シミュレーションを行った。ここでは想定した検出器で検出した散乱回数ごとの光子の分布、エネルギースペクトル、シミュレーションの計算時間を結果として得る。想定したジオメトリとシミュレーション条件を以下に示す。

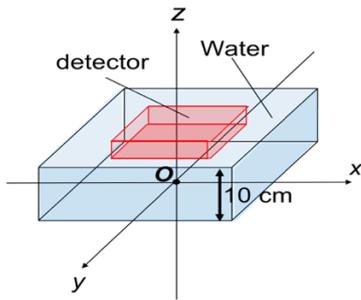


図6 シミュレーションのジオメトリ

表1 シミュレーション条件

媒質	H ₂ O Al
発生光子数	1億個
光子発生位置	原点
光子放出方向	z軸原点方向
初期光子エネルギー	30 keV
最大散乱回数	5回
検出器サイズ	32.5 × 32.5 cm ²
検出器のピクセルサイズ	0.5 × 0.5 cm ²

また、EGS5の実行環境 (CPU) と自作コード (GPU) の実行環境を以下に示す。

表2 実行環境 (CPU)

CPU	Kabylake-Y
クロック周波数	1.2 GHz
メモリ	8 GB

表3 実行環境 (GPU)

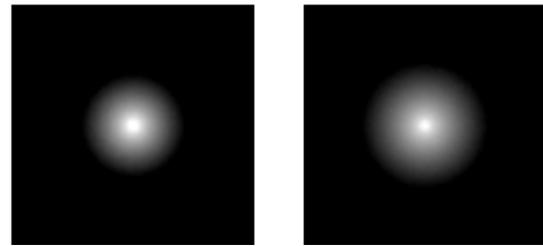
GPU	Tesla P100(Pascal)
Grid Size	100,00
Block Size	1,000

c) シミュレーション結果

自作コード, EGS のシミュレーション結果を以下に示す。まず、自作コードによるシミュレーションで検出された光子の分布を以下に示す。



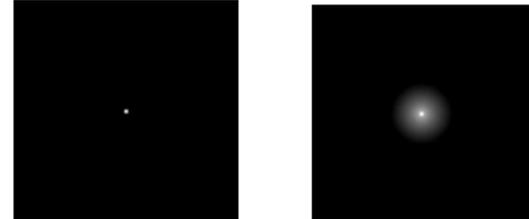
(a) 散乱回数0回 (b) 散乱回数1回



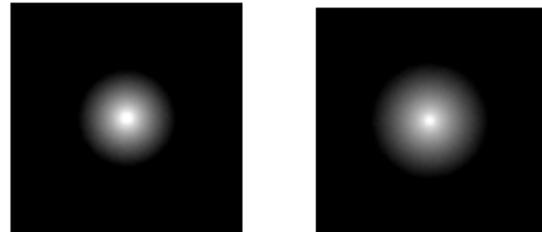
(c) 散乱回数3回 (d) 散乱回数5回

図7 検出された光子の分布 (自作コード)

次に EGS によるシミュレーションで検出された光子の分布を示す。



(a) 散乱回数0回 (b) 散乱回数1回



(c) 散乱回数3回 (d) 散乱回数5回

図8 検出された光子の分布 (EGS5)

さらに、上図の中心から、x軸方向に切ったプロファイルの比較を以下に示す。

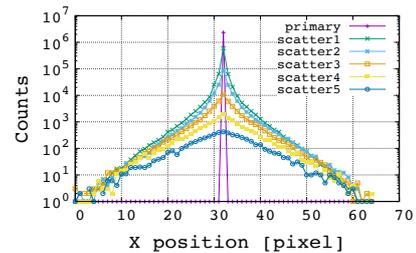


図9 図7のプロファイル (自作コード)

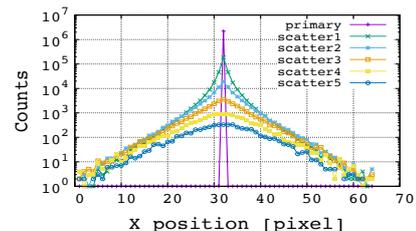


図10 図8のプロファイル (EGS5)

次に自作コードと EGS5 それぞれでシミュレーションした際の実行時間の比較を以下に示す。

表 4 実行時間の比較

自作コード (GPU)	124 s
EGS5 (CPU)	2867 s

自作コードの実行時間 124 s は、EGS5 の 2867 s に比べ、約 23 倍高速になった。

また、GPU の評価のために、ブロックサイズ (ブロックあたりのスレッド数) を変更して実行時間を測定した結果を以下に示す。

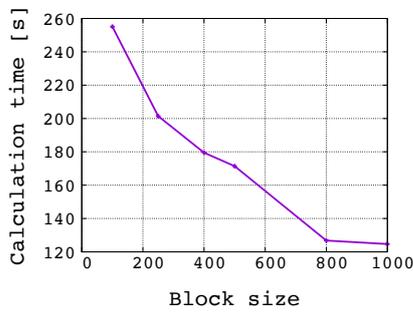


図 11 ブロックサイズと実行時間の関係

上図からブロックサイズが増すにつれ、実行時間が短くなっていることがわかる。

(2) 散乱線の定量評価

自作コードの試用として、マンモグラフィ実機に即した条件で、散乱線の定量評価を行った

a) シミュレーション方法

ジオメトリはマンモグラフィの実機に合わせ、線源には管電圧 30 kV の X 線を用いた。検出器上に水、カルシウム球を含む二媒質のファントムを想定した。

以下に、にジオメトリを示す。

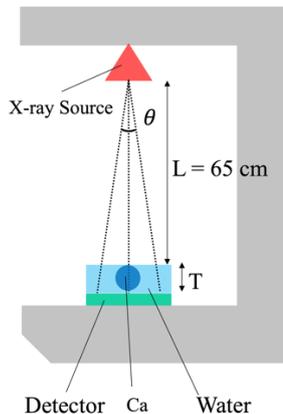


図 12 シミュレーションのジオメトリ

また、図 2 に示した X 線を線源として設定した。次に、シミュレーション条件を示す。

表 5 シミュレーション条件

ターゲット	Mo
フィルタ	Mo(30 μm)
検出器	20 \times 20 cm^2
検出器のピクセルサイズ	0.02 \times 0.02 cm^2

以上の条件で、水の厚み (カルシウムなし)、カルシウム球の半径、X 線の入射角を変更した際の散乱線率を測定した。

b) シミュレーション結果

散乱線の定量評価の結果を示す。散乱線の評価には散乱線率 (= 散乱光子数 / プライマリ光子数) を用いる。まず、水の厚みと散乱線率の関係を以下に示す。

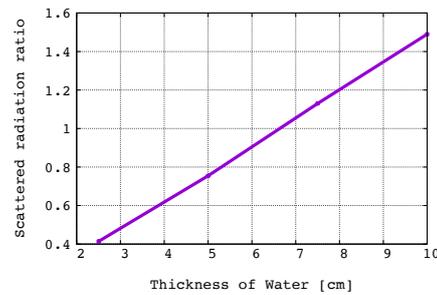


図 13 水の厚みと散乱線率の関係

水の厚みが大きくなるほど、散乱線の割合が多くなることがわかる。

次に、カルシウム球の半径と散乱線率の関係を以下、に示す。

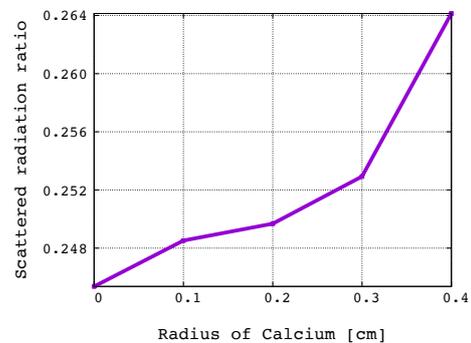


図 14 カルシウム球の半径と散乱線率の関係

カルシウム球の半径が大きくなるほど、散乱線の割合が多くなることがわかる。

次に、X 線の入射角と散乱線率の関係を以下に示す。

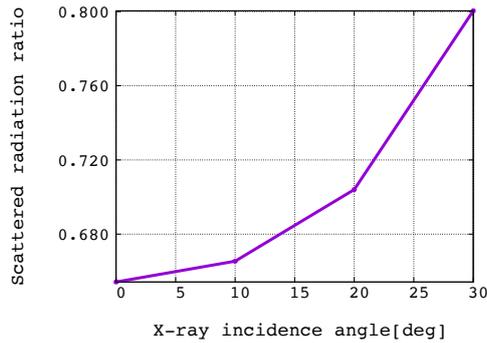


図 15 X 線入射角と散乱線率の関係

X 線の入射角度が大きくなるほど、散乱線の割合が多くなることがわかる。

6. 機械学習を用いた散乱線除去

GPU を用いた並列計算ができる自作コードにより、光子輸送シミュレーションが高速で実行できるようになったため、これを機械学習のデータの取得に応用した。本研究では、U-net を用いて、自作コードにより取得したデータを学習し、散乱線の除去を試みた。

(1) U-net

U-Net は、医用画像のセマンティックセグメンテーション向けに提案された、CNN 構造である。以下に U-net の構造を示す。

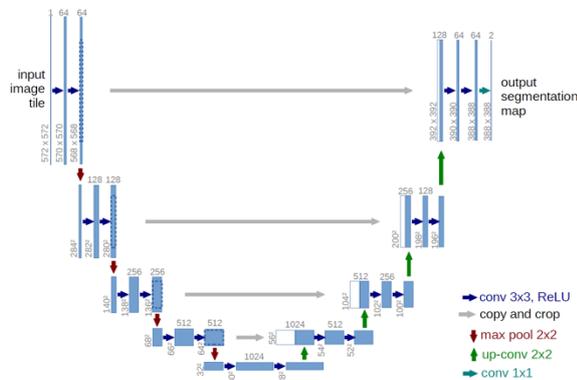


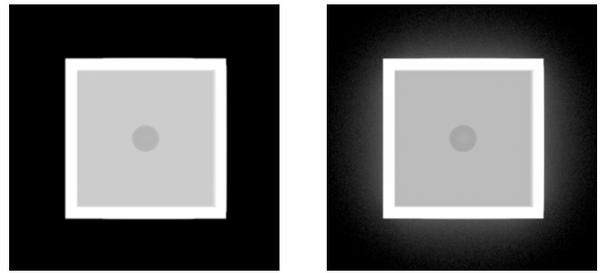
図 16 U-net の構造[4]

FCN(Fully Convolutional Network) をもとに、後半 Decoder 部分をプーリング層から(学習可能な)アップサンプリング層に改善した構造であり、各層は、アルファベットの「U」の形をした Encoder-Decoder 構成の対称型のネットワークになっている。

(2) データセット

自作コードにより、学習用のデータセットを取得した。図 12、表 5 の条件で、X 線入射角やカルシウム厚などを変更してさまざまな画像を作成した。

以下に取得したデータセットの例を示す。



(a) 教師画像 (b) 入力画像

図 17 データセットの例

左の教師データはプライマリ成分のみ含まれている画像であり、右の入力データはプライマリ成分に加え、散乱線成分も含まれている。また、学習のために画素値の範囲を最小値 0、最大値 1 に正規化したものである。次に、上図に示した、画像のプロファイルの比較を以下に示す。

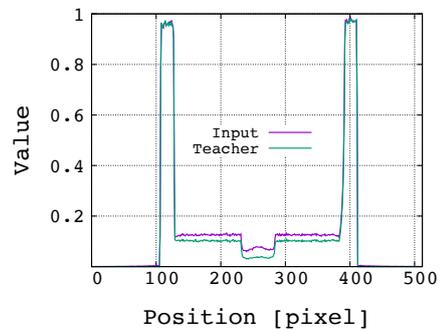
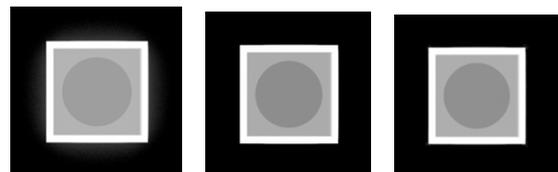


図 18 データセットのプロファイル

以上のような画像をデータセットとし、学習を行った。今回の研究では自作コードで作成したデータが散乱線除去のネットワークに活用できることの確認のため、データセットは 10 組、20 枚で行った。そのため過学習になってしまうことは前提とし、epoch 数 3000、バッチサイズ 10 で行った。

(3) 結果

まず、学習の確認をするために訓練の結果を出力したものを、以下に示す。



(a) 入力画像 (b) 教師画像 (c) 出力画像

図 19 訓練結果画像

以下にプロファイルを示す。

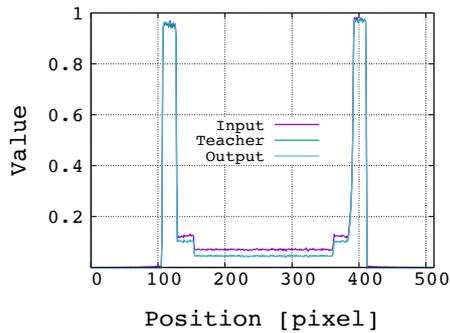


図 20 図 19 のプロファイルの比較

次に、テストの結果を出力したものを、以下に示す。

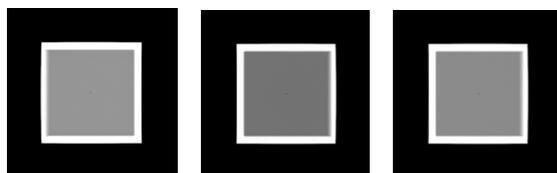


図 21 テスト結果画像

以下にプロファイルを示す。

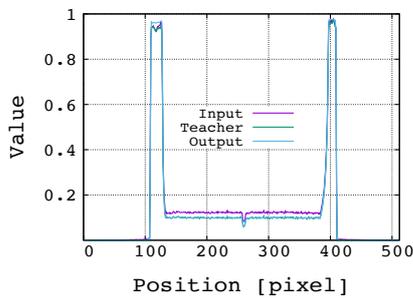


図 22 図 21 のプロファイルの比較

プロファイルを見ると、散乱線成分が除去され、教師データに近づいたことがわかる。次に学習時の教師画像と学習結果画像の差 (MSE) をプロットした損失関数を以下に示す。

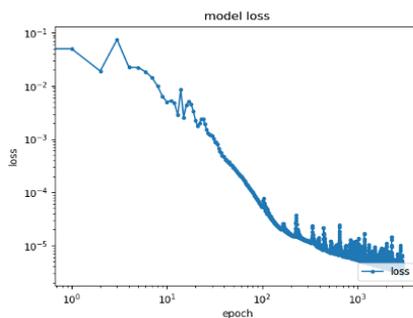
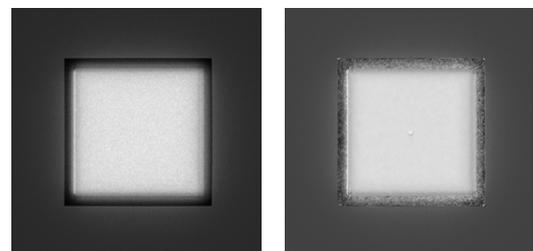


図 23 損失関数

1000epoch あたりから、グラフが横ばいになり始めている。

また、散乱線除去ができていないかの確認、すなわち、散乱線成分を正しく推定できているかの確認のために、散乱線成分を比較した。入力画像には散乱線成分が含まれ、教師画像には散乱線成分が含まれない。つまり、実際の散乱線成分は、入力画像 - 教師画像、ネットワークでの推定散乱線成分は、入力画像 - 出力画像で表すことができる。

テストデータの 2 つの散乱線成分を画像化したものと、そのプロファイルを以下に示す。



(a) 実際の散乱線成分 (b) 推定散乱線成分

図 23 実際の散乱線と推定散乱線の比較

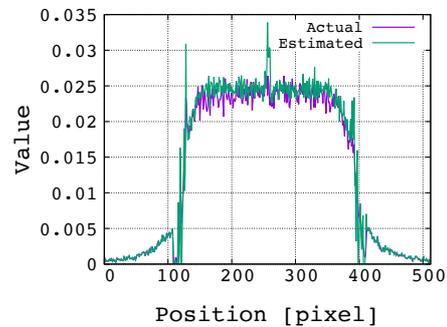


図 24 図 23 のプロファイルの比較

カルシウムと水、空気と水のような媒質の境目あたりで、推定散乱線の値が大きくなってしまっている部分があるが、概形は概ね一致している。

7. 考察

(1) EGS との比較について

まず、図 9 と図 10 の 2 つグラフを比較すると、散乱回数ごとの検出された光子の分布の概形が概ね一致している。ここで、プライマリ光子数について考える。プライマリ光子数は 1 回も散乱せずに検出器に到達した光子の数であり、本シミュレーションでは単一エネルギーの線源を設定しているため、Lambert Beer の法則により理論値を求められる。算出された光子の理論値は 2, 328, 374 個

であり、自作コードによるシミュレーションで検出されたプライマリ光子数は 2,326,869 である。これは期待値 2328374 ± 1526 の範囲内であるため、妥当な値であると言える。概形の概ね一致、プライマリ光子の理論値より、自作コードは EGS5 と同等の精度でのシミュレーションを行うことができたと考える。また、ブロックサイズと実行時間の比較については、図 11 から、ブロックサイズの増加に伴い、実行時間が短くなっているのがわかる。ブロックサイズとは、1 ブロック内に含むスレッドの数であり、CUDA のカーネル関数は、1 ブロック内のスレッドを並列処理させるので大きくするほど並列化が効率的になるからであると考えられる。今回は光子 1 つ 1 つの挙動計算を並列化したため、ブロックサイズ×グリッドサイズ(総スレッド数)を総光子数 1 億個に設定した。よって、CUDA の仕様である上限の 1024 以下で、1 億のスレッド数を割り切れる最も大きな値として、ブロックサイズ 1000 を設定したのは、最も効率的なパラメータであると考えられる。

(2) 散乱線の定量評価について

図 13, 15 より、それぞれ、水ファントムの厚み、X 線入射角の増加に伴って、散乱線率が大きくなっているのがわかる。これは水の厚み、入射角が大きくなるにつれ、光子が媒質内を通過する距離が長くなり、散乱が起こる確率が上がるため、水の厚み、入射角の増加にともなって散乱線率が上がるのは妥当である。図 14 ではカルシウム級の半径の増加に伴い散乱線率が増加しているが、水に比べカルシウムの減衰係数が大きい、つまり、相互作用の起こる確率が大きい、カルシウムの厚みの増加にともなって散乱線率が上がるのは妥当である。

(3) 機械学習による散乱線推定について

テストデータ、訓練データの結果画像をそれぞれ比較すると、わずかであるが、入力画像に見られる白くぼやけていたような散乱線成分が、出力画像では取り除かれていることがわかる。また、数値による考察のために、MSE を計算した結果、種訓練データの、入力画像と教師画像の差から求めた MSE は $1.356603e-4$ 、出力画像と教師画像の差から求めた MSE は $4.494453e-06$ である。実際の散乱線成分を表す入力画像と教師画像の MSE に対し、出力画像と教師画像の MSE は 3.3%程度まで小さくなっており、正しく散乱線を推定できたと考えられる。

8. 結論

本研究では、マンモグラフィ画像の散乱線のために、GPU で並列処理可能な光子輸送シミュレーションの高速化コードを開発した。自作コードの EGS の精度と計算時間の比較とし、簡易的なシミュレーションを行った結果、自作コードは EGS と同程度の精度であることが確認でき、計算時間においては 23 倍の高速化を図ることができた。また、この自作コードを用いた光子輸送計算でマンモグラフィのシミュレーションを実施し、散乱線の定量評価および、U-net による散乱線推定を行った。入力画像に

含まれていた散乱線成分を機械学習により、約 3.3%の MSE で正しく推定することができた。以上により、自作コードによるマンモグラフィの画像取得シミュレーションを高速で行うことができ、そのデータを用いて機械学習による散乱線の除去が可能となった。

謝辞

本研究にあたり、全般においてご指導していただきました尾川浩一教授、様々な面でお世話になりました村田一心氏、マンモグラフィの実機データを提供していただきました乳井嘉之氏、機械学習のネットワークについての助言をしていただきました島田良氏、および尾川研究室の方々に深く感謝いたします。

参考文献

- 1) T. Tanaka, T. Kurosawa, and N. Saito, "Dose standard for safe and secure breast cancer examination", *Synthesiology* Vol.5 No.4 pp.222-223 (Nov.2012)
- 2) H. Hirayama, Y. Namito, A. F. Bielajew, S. J. Wilderman, W. R. Nelson, "THE EGS5 CODE SYSTEM," January 13, 2016.
- 3) J. H. Hubbell, W. J. Veigele, E. A. Briggs, R. T. Brown, D. T. Cromer, and R. J. Howerton, "Atomic form factors, incoherent scattering functions, and photon scattering cross sections," *J. Phys. Chem. Ref. Data*, 471-538, 1975.
- 4) O. Ronneberger, P. Fischer, T. Brox "U-Net: Convolutional Networks for Biomedical Image Segmentation" 1,2 (2015)