

グラフィックスハードウェアを用いた放射型 CT 画像再構成の高速化 —再投影と逆投影のシステム行列が一致する新しい実装—

無所属*¹ / 株式会社島津製作所*² /

筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻*³

李 美花*¹/ 赤澤礼子*²/ 工藤博幸*³/ 北村圭司*²

はじめに

放射型 CT 装置では、統計雑音・ γ 線吸収・散乱線などの画質劣化要因が存在するため、それらの影響を考慮した統計的推定に基づく反復法が画像再構成に用いられる。統計的推定に基づく反復法としては、代表的なものに MLEM 法・OSEM 法・RAMLA 法・DRAMA 法とその一般化である DOSEM (Dynamic OSEM) 法がある¹⁾⁵⁾。

本論文では、NVIDIA 社の GPU (Graphic Processing Unit) である GeForce 8800 GTS512 とその開発環境である CUDA (Compute Unified Device Architecture) アーキテクチャを用いて、DOSEM 法 (DRAMA 法は 1 つのサブセット内に 1 方向の投影データのみが含まれる DOSEM 法の特別な場合) の高速化を試みた。ここで、DRAMA 法ではなく DOSEM 法を採用したのは、以下の理由のためである。再投影演算を高速化する場合、並列化は複数の投影データに対する処理を同時に行うことで実現する。一方、逆投影演算を高速化する場合、複数の画素に対する処理を同時に行うことで実現する。DOSEM 法の方が DRAMA 法より各サブセットに含まれる投影データ方向数が多いため、再投影における並列化のレベルを RAMLA 法や DRAMA 法よりも大きくし、GPU の並列化機能をより有効に利用することができる。

本論文では、従来研究である文献 6)-8) とは異なり、逆投影で使用される Pixel-driven のシステム行列が再投影で使用される Ray-driven のシステム行列と全く等価になるように、逆投影のアルゴリズムを改良し、256 方向×256 点のサイノグラムから 256×256 画素の 2 次元画像を再構成する実験を行った。実験の結果、DOSEM 法のサブセット数が 8 の場合で一反復の計算時間は 3.0ms であり、CPU (Intel Core 2 Duo E6320 1.86GHz, 2GB RAM) で計算を行う場合の約 129 倍高速化された。また、再構成画像の画質は、GeForce GTS512 が浮動小数点演算を行うため、CPU で実装した場合と全く同じであった。

CUDA および DOSEM 法

1. CUDA アーキテクチャ

CUDA アーキテクチャは、C 言語を用いて NVIDIA 社の GeForce 8800 シリーズ及びそれ

以降の GPU を汎用計算の並列プロセッサとして活用するための開発環境である。CUDA アーキテクチャでは、図 1 に示すように汎用計算を CPU とカーネル関数の間で直接行うことができ、OpenGL や Direct3D のような GPU グラフィックスのパイプライン方式でプログラムを書く必要がない利点がある。CUDA 開発環境において、カーネル関数を実行すると、GPU は多数の Thread (カーネル関数に呼び出される処理) を並列に実行する。

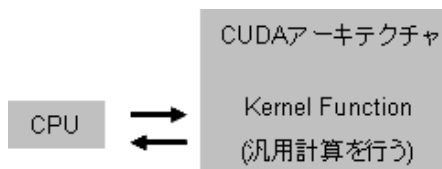


図 1 CUDA アーキテクチャ.

2. DOSEM 法

DOSEM 法は Dynamic OSEM 法の略で、その反復式は次のように表される。

$$x_j^{(n,m+1)} = x_j^{(n,m)} \left[1 + \frac{\lambda^{(n,m)}}{C_j} \sum_{i \in S_m} a_{ij} \left(\frac{y_i}{\sum_{k=1}^J a_{ik} x_k^{(n,m)}} - 1 \right) \right] \quad (1)$$

$$x_j^{(n+1,0)} = x_j^{(n,M)} \quad (2)$$

ただし、 x_j は画素 j から放出される光子数の平均、 y_i は測定された i 番目の投影データ、 a_{ij} は x_j と y_i を関係づけるシステム行列の要素を表す。また、 n はメイン反復回数、 M はサブセット数、 m はサブセット番号、 S_m は m 番目のサブセットに属する投影データインデックスの集合を表す。式(1)において、正規化行列 C_j 及び緩和係数 $\lambda^{(n,m)}$ は次のように定義される。

$$C_j = \max_m \sum_{i \in S_m} a_{ij} \quad (3)$$

$$\lambda^{(n,m)} = \beta_0 / (\beta_0 + m + \gamma M), \quad 0 \leq \gamma \leq 1 \quad (4)$$

ただし、式(4)に含まれるパラメータ β_0 はサブセット間の雑音の不均衡を抑える役割をもち、文献 4) で導出された手法により計算できる。

DOSEM 法において、再投影は $y_i^{(n,m)} = y_i / \sum_{k=1}^J a_{ik} x_k^{(n,m)} - 1, i \in S_m$ を計算する部分であり、逆投影

は $x_j^{(n,m+1)} = x_j^{(n,m)} (1 + \frac{\lambda^{(n,m)}}{C_j} \sum_{i \in S_m} a_{ij} y_i^{(n,m)})$, $j=1, \dots, J$ を計算する部分である。DOSEM 法の実装において、

多くの時間を要する計算は、次の再投影と逆投影の和を計算する部分である。

$$\sum_{k=1}^J a_{ik} x_k^{(n,m)}, \quad i \in S_m \quad (5)$$

$$\sum_{i \in S_m} a_{ij} y_i^{(n,m)}, \quad j = 1, \dots, J \tag{6}$$

提案手法

従来の GPU を用いて式(5), (6)を高速化する研究では, グラフィックスパイプラインを用いて処理を行うため, Texture mapping 技術で再投影と逆投影を計算している^{6),7)}. そのため, 再投影において使用されるシステム行列と逆投影において使用されるシステム行列が一致しない問題点がある*. システム行列の不一致は, 再構成画像の画質や反復法の収束性に悪影響を及ぼすことが知られている. 本章では, CUDA を用いて再投影と逆投影のシステム行列を一致できる DOSEM 法の高速な実装方法について説明する. 以降では, まず再投影と逆投影の計算方法について説明し, 次に実装全体の流れを説明する.

1. 再投影の計算

式(5)の再投影部分を高速に並列化するため, 変数*i*について並列化を行いサブセット S_m に含まれる多数の再投影データ $y_i^{(n,m)}$ ($i \in S_m$) を同時に計算する. また, 各再投影データ $y_i^{(n,m)}$ の計算には, 図 2 に示す標準的な Joseph の再投影アルゴリズムを用いる⁹⁾. Joseph の再投影アルゴリズムは, 投影データ y_i に対応する直線に対して, 変化が大きい方向を主方向とし, 変化が小さい方向を補間に使用する. そのため, 画素の離散化に起因するエリアシング誤差を低減する効果がある.

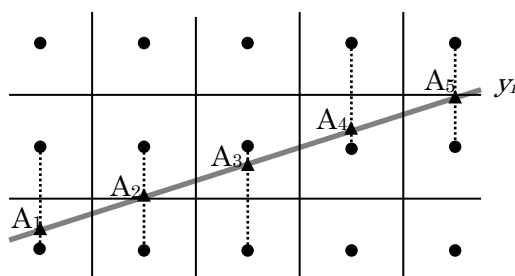


図 2 Joseph の再投影法.

DOSEM 法の再投影部分を計算する擬似コードは, 以下のようになる.

```
__global__ void reprojection_cuda(int m, int na, float* d_sg, float* d_y)
{
    int tp = blockIdx.x*blockDim.x+threadIdx.x;
```

* 仮に, 逆投影も Ray-driven で並列化できたとしても, 複数の Thread が画像配列にデータを同時書き込みする必要が生じ, 並列化の効率が低下する.

```

int ts = blockIdx.y*blockDim.y+threadIdx.y;
float P=0.0;
Q=d_sg[m*na*NP+ts*NP+tp];
for(int i=0;i<NR;i++){
    P+=tex2D(tex, i+0.5, pos(i+0.5)); あるいは P+=tex2D(tex, pos(i+0.5), i+0.5);
}
d_y[ts*NP+tp]=Q/P-1.0;
}

```

ただし、擬似コードにおいて、 m はサブセット番号、 na はサブセット内の投影データ方向数、 d_sg は DOSEM 法における投影データアクセス順序に合うようにデータの順番を並べ替えたサイノグラム、 d_y は計算される再投影結果の値 $y_i^{(n,m)}$ である。また、 ts はサブセット内の投影方向番号、 tp は 1 つの投影方向に含まれる投影データの番号、 NP は 1 つの方向内の投影データ数、 NR は画像の各座標方向の画素数、 tex は画像の値とリンクするテクスチャメモリである。CUDA 関数 $tex2D(\cdot)$ は、GPU のテクスチャハードウェアで高速に補間演算を行う関数であり、 $pos_{(i+0.5)}$ は点 $(i+0.5, pos_{(i+0.5)})$ が投影データ y_i に対応する直線上にくるように求めた値である。各定数の間には $gridDim.x \times blockDim.x = NP$, $gridDim.y \times blockDim.y = na$ の関係がある。

2. 逆投影の計算

式(6)の逆投影部分を高速に並列化するため、変数 j について並列化を行い多数の画素に対する逆投影結果 $x_j^{(n,m+1)}$, $j \in (1, 2, \dots, J)$ を同時に計算する。また、従来の GPU を用いた実装の問題点である再投影と逆投影におけるシステム行列の不一致を避けるため、Ray-driven の再投影と全く等価なシステム行列を使用できる図 3 に示す手法を用いて逆投影を計算する。この手法では、まず各画素値 x_j に寄与する投影データのインデックス集合 $i, i+1$ (サイノグラムの場合は各投影方向あたり 2 つ存在する) を求め、得られたインデックスに対して Joseph の再投影アルゴリズムと全く同じ $a_{ij}, a_{i+1,j}$ を直接計算する。具体的な数式で表すと、画素 x_j に対応する 2 次元座標を (t_i, t_j) とする場合、インデックス i から a_{ij} を求める計算は次式のようなになる (インデックス $i+1$ から $a_{i+1,j}$ を求める計算も同様である)。

$$a_{ij} = \frac{i+0.5+(t_i+0.5)\sin\theta}{\cos\theta} - (t_j-0.5) \quad (7)$$

この実装方法は、余分なメモリへの書き込みと読み込みがないため、システム行列不一致の問題点を解決する有効な手法であると考えられる。

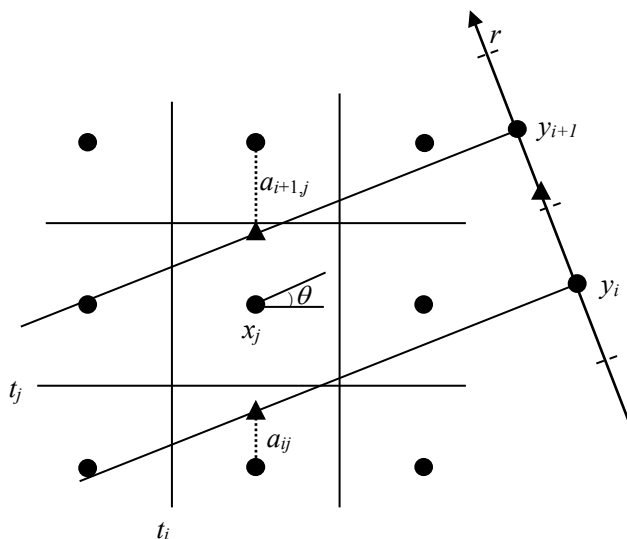


図3 システム行列が Joseph の再投影法にマッチングするように計算する新しい逆投影手法.

DOSEM 法の逆投影部分を計算する擬似コードは、以下のようになる.

```

__global__ void backprojection_cuda(int m, int na, float lambda, float* Cj, float* d_y, float* d_xk)
{
    int ti = blockIdx.x*blockDim.x+threadIdx.x;
    int tj = blockIdx.y*blockDim.y+threadIdx.y;
    float ay=0.0;
    for(int angle=0; angle<na; angle++)
    {
        (1)  $\tan(\theta(m * na + \text{angle}))$  の値で、主方向が水平方向か垂直方向を決定
        (2) 画素(ti, tj)から投影座標 i,i+1 を計算
        (3) 式(7)により  $a_{ij}, a_{i+1,j}$  を計算
        (4)  $ay += a_{i+1,j}*d_y[i+1]+a_{ij}*d_y[i];$ 
    }
    d_xk[tj*NR+ti] *= lambda*ay/Cj [tj*NR+ti]+1.0;
}

```

ただし、上述の擬似コードにおいて、 λ は DOSEM 法の緩和係数、 C_j は式(3)により計算された正規化行列、 d_xk は式(6)により計算される逆投影結果の値 $x_j^{(n,m+1)}$ 、 t_i, t_j は画像の水平・垂直方向の画素番号を表す.

3. 実装全体の流れ

提案手法に基づく実装全体の流れをまとめると、以下のようになる。

[STEP1] CPU 側で、サイノグラム $y_i (i=1,2,\dots,J)$ のデータ順序を DOSEM 法における投影データアクセス順序に合うように並べ替え、初期画像 $x_j^{(0,0)} (j=1,2,\dots,J)$ として全画素で同じ値をもつ平らな画像を与える。

[STEP2] 並べ替えたサイノグラムと初期画像を、GPU のメモリに移動する。

[STEP3] 各反復回数 $n=0,1,\dots,N-1$ において、サブセット $m=0,1,\dots,M-1$ 毎に本章の 1. と 2. で説明した再投影を行うカーネル関数と逆投影を行うカーネル関数を実行する。

[STEP4] 反復が終わったら、GPU 側にある再構成結果 $x_j^{(N,0)} (j=1,2,\dots,J)$ を CPU にコピーする。

実験結果

1. 高速性

GeForce 8800 GTS512 と CUDA 2.0 を用いて実験を行った結果、光子の全計数値を 1,000 万個とするポアソン雑音を加えた 256 方向×256 点のサイノグラム形式の投影データから対象物体を再構成する場合、サブセット数が 8 の DOSEM 法の一反復当たりの計算時間は 3ms であった^{**}。この計算時間は CPU (Intel Core 2 Duo E6320 1.86GHz, 2GB RAM) で実装した場合の 387.3ms と比較して、約 129 倍高速であった。また、再構成画像の画質は GeForce 8800 GTS512 が浮動小数点演算を行うため、CPU で実装した場合と全く同じであった。図 4 に、実験に用いた Shepp-Logan ファントムおよび GPU と CPU で再構成した画像を示す。

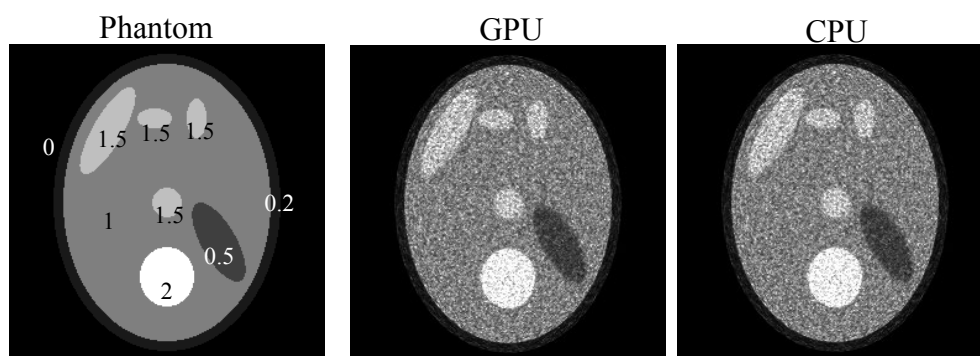


図 4 GPU と CPU により再構成した画像。

^{**} 同じプログラムを NVidia 社の異なる GPU モデルである TESLA C870 で実行した場合の計算時間は 3.4ms, Geforce 8800 GT で実行した場合の計算時間は 3.6ms であった。

2. システム行列不一致の影響

本論文で提案したシステム行列不一致の問題点を解決する新しい Pixel-driven 逆投影アルゴリズムと従来の Pixel-driven アルゴリズムの相違を, DOSEM 法で評価した. その結果, 新しい手法による画質の改善は見られたが両者の相違は顕著でなかった. その理由は, DOSEM 法に収束性が良くシステム行列不一致の影響を抑える効果が元々あるためと考えられる. そこで, 収束性が悪くシステム行列不一致の影響が大きいと予想される OSEM 法についても, DOSEM 法と同じように実装を行った. 図 5 に雑音が含まれていない投影データからの再構成画像を示す. この結果から, 特に OSEM 法のように収束性が悪い反復法では, システム行列不一致は再構成画像に大きなアーチファクトを生じさせることが分かる.

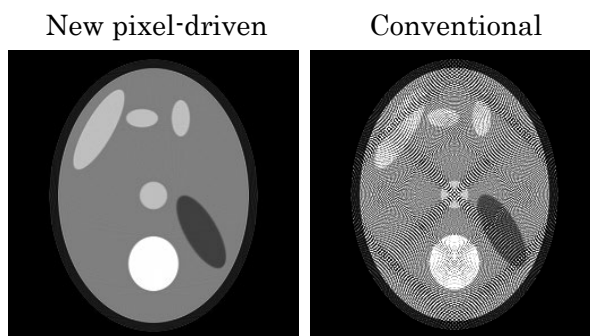


図5 OSEM法により再構成した画像.

おわりに

本論文では, NVidia 社の GPU である GeForce 8800 GTS 512 と CUDA アーキテクチャを用いて, 放射型 CT の画像再構成法である DOSEM 法を高速化することを試みた. 256 方向×256 点のサイノグラムから 256×256 画素の 2 次元画像を再構成する実験を行った結果, DOSEM 法のサブセット数が 8 の場合で一反復あたりの計算時間は 3.0ms であり, CPU で計算を行う場合と比較して約 129 倍高速化された. また, GeForce 8800 GTS512 は浮動小数点演算を行うため, 再構成画像の画質は CPU で実装した場合と全く同じであった. 更に, 実装の過程において, 従来の GPU を用いた反復法の実装で問題となっている再投影と逆投影におけるシステム行列の不一致を避けるため, Ray-driven の再投影と全く等価なシステム行列を使用できる Pixel-driven の逆投影アルゴリズムを提案した.

今後の課題は, GPU と CUDA を用いて, 放射型 CT の 3 次元や 4 次元画像再構成のための DOSEM 法の高速度プログラムを実装することである. サイズの大きい 3 次元や 4 次元再構成問題では, 2 次元と比較してより大きな高速化が期待される.

文献

- 1) Hudson HM and Larkin RS: Accelerated image reconstruction using ordered subsets of projection data. *IEEE Trans Med Imaging*, 13: 601-609, 1994
- 2) Browne J and De Pierro AR: A row-action alternative to the EM algorithm for maximizing likelihoods in emission tomography. *IEEE Trans Med Imaging*, 15: 687-699, 1996
- 3) 田中栄一 : PET 画像の再構成法の現状と展望. *日本放射線技術学会雑誌* 62: 771-777, 2006
- 4) Tanaka E, Kudo H: Subset-dependent relaxation in block-iterative algorithm for image reconstruction in emission tomography. *Phys Med Biol* 48: 1405-1422, 2003
- 5) Tanaka E, Kudo H: Acceleration of statistical image reconstruction for transmission CT using DRAMA (dynamic RAMLA) strategy. *Med Imag Tech* 24: 114-124, 2006
- 6) Xu F, Mueller K: Accelerating popular tomographic reconstruction on commodity PC graphics hardware. *IEEE Trans Nucl Sci* 52: 654-663, 2005
- 7) Kole J S, Beekman F J: Evaluation of accelerated iterative x-ray CT image reconstruction using floating point graphics hardware. *Phys Med Biol* 51: 875-889, 2006
- 8) Pratz G, Chinn G, Habte F et al: Acceleration of fully 3-D list-mode OSEM for high-resolution PET using graphics processing units, *Proc Fully 3D Meeting and HPIR Workshop*: 41-44, 2007
- 9) Joseph P M: An improved algorithm for reprojecting rays through pixel images, *IEEE Trans Med Imaging* 1: 193-196, 1982