

断層映像法の基礎 第32回 ML-EM法とOS-EM法

篠原 広行¹⁾、桑山 潤¹⁾、小川 亙¹⁾、橋本 雄幸²⁾

¹⁾ 首都大学東京人間健康科学研究科 放射線科学域

²⁾ 横浜創英短期大学 情報学科

はじめに

第31回では繰り返しを利用して徐々に解に近づけていく方法を紹介した。本稿ではその繰り返しを使った方法で最も多く使われているML-EM法とOS-EM法について解説する。また、その方法を利用した数値シミュレーションの結果についても紹介する。

1. 投影、逆投影と検出確率
2. ML-EM法
3. OS-EM法

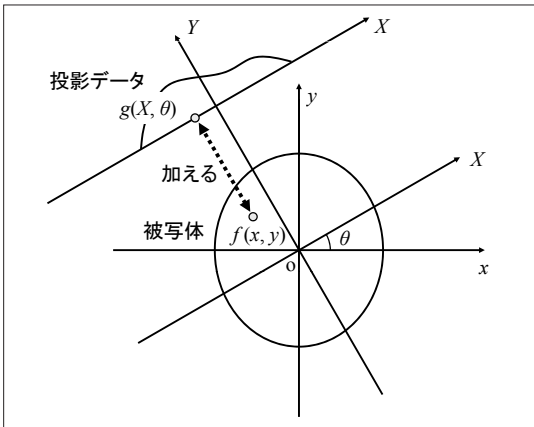


図1. 投影と逆投影の関係

投影では、被写体の値を対応する投影データに加え、逆投影では投影データの値を対応する被写体の位置に加える。

1. 投影、逆投影と検出確率

繰り返しの方法を用いるときに投影と逆投影を利用することは、前回のART法とSIRT法で紹介している。ここで投影と逆投影について改めて考えてみる。

まずは数式で比較すると、投影の式は、

$$g(X, \theta) = \int_{-\infty}^{\infty} f(x, y) dY \quad (1)$$

となり、逆投影の式は、

$$f'(x, y) = \frac{1}{2\pi} \int_0^{2\pi} g(X, \theta) d\theta \quad (2)$$

となる。投影は、被写体の座標 (x, y) の値を Y 軸に沿って対応する投影データの (X, θ) の位置に加える。逆投影は、投影データの (X, θ) の値を被写体の座標 (x, y) の位置に加える。これは図1に示すように、両者の位置関係が同じであることがいえる。ただし、両者ともに加える操作を行うので、値が元に戻るわけではない。

この位置関係をデジタルデータで考える。デジタルデータでは被写体はデジタル画像 $f(x_j, y_i)$ となり、投影データはある角度の個々の検出器 $g(X_n, \theta_m)$ となる。これらの関係は図2のように表される。被写体である画像も投影データも離散化されているので、画像の画素の座標と対応する投影データの1つの検出器位置は、ぴったり合わずにずれてしまうことの方が多い。デジタルデータの場合、位置がぴったり合わないことはよく起こることである。その場合は、近くの値から補間という操作で値を推定する

連絡先： ☎ 116-8551

東京都荒川区東尾久 7-2-10

首都大学東京人間健康科学研究科放射線科学域 篠原 広行

TEL：03-3819-1211 FAX：03-3819-1406

E-mail：sinohara@hs.tmu.ac.jp

URL：http://www.metro-hs.ac.jp/rs/sinohara

ことで対処している。通常、投影と逆投影の操作は、別々の補間を行って実行されるが、位置の対応関係は同じなので、1つにまとめることができる。そもそも補間という処理は、位置の対応がぴったり一致しないときに、周りの値から寄与する割合を求める処理である。位置の対応関係が同じであれば、お互いの関係し合う割合を決めておくことができる。

投影と逆投影では、画像の画素と投影データの1つの検出器位置との関係が対象になる。具体的には図3に示すように、1つの画素と1つの検出器が関係し合う割合を計算する。最も単純な計算方法は、画素の幅と検出器の幅を1にして、画素が対応する検出器に見込む面積を求めて、それを割合とする方法である。1つの画素の値を複数の検出器に分け与える考え方である。この割合を検出確率と呼ぶ。この検出確率は、すべての画素からすべての検出器に

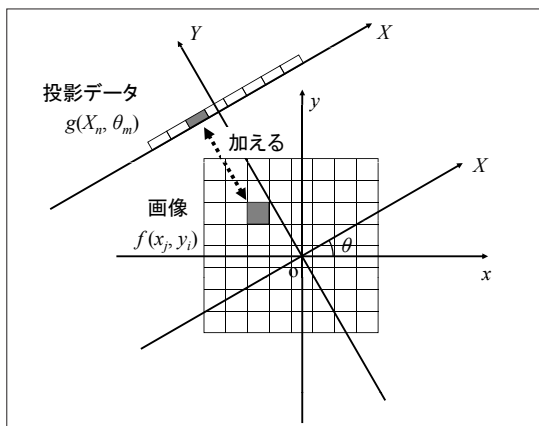


図2. デジタルデータにおける投影と逆投影の関係
デジタルデータでは、画素と検出器との関係になる。

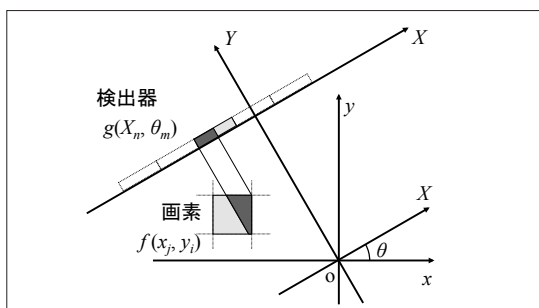


図3. 1つの画素と1つの検出器との関係
1つの画素が1つの検出器にどのくらい見込めるかの割合が検出確率になる。

対して計算する。画素 (x_j, y_i) と検出器 (X_n, θ_m) との関係になるので、検出確率は $(x_j, y_i, X_n, \theta_m)$ の関数となる。検出確率を $C(x_j, y_i, X_n, \theta_m)$ と置くと、投影の式は、

$$g(X_n, \theta_m) = \sum_{i=0}^{My-Mx-1} \sum_{j=0}^{Mx-1} C(x_j, y_i, X_n, \theta_m) f(x_j, y_i) \quad (3)$$

となり、逆投影の式は、

$$f'(x_j, y_i) = \frac{1}{\sum_{n=0}^{Np-1} \sum_{m=0}^{Na-1} C(x_j, y_i, X_n, \theta_m)} \sum_{n=0}^{Np-1} \sum_{m=0}^{Na-1} C(x_j, y_i, X_n, \theta_m) g(X_n, \theta_m) \quad (4)$$

となる。ここで、 Mx と My はそれぞれ画像の幅と高さに相当し、 Np と Na はそれぞれ投影データの1投影あたりの検出器の数と投影数に相当する。また逆投影に分数がついているのは、検出確率を画素から検出器に見込んで算出しているの、画素に全方向から逆投影される全確率を1にするための規格化の計算である。

変数の次元が多くなると見づらくなるので、画素と検出器を1次元変数に直して表記すると、投影の式は、

$$y_i = \sum_{j=0}^{M-1} C_{ij} \lambda_j \quad (5)$$

となり、逆投影の式は、

$$\lambda'_j = \frac{1}{\sum_{i=0}^{N-1} C_{ij}} \sum_{i=0}^{N-1} y_i C_{ij} \quad (6)$$

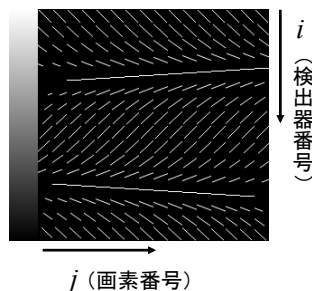


図4. 16×16画素の画像と16×16投影の投影データから作成した検出確率 C_{ij} の画像 (256×256マトリクス)
横方向が画像の画素番号に相当し、縦方向が投影の検出器番号に相当する。

となる。ここで、 λ_j と λ'_j は画像を表し、 j は画素の番号に相当する。 M は画素の総数である。 y_i は投影データを表し、 i は検出器の番号に相当する。 N は検出器の総数である。 C_{ij} は i 番目の検出器と j 番目の画素との検出確率を表す。

検出確率 C_{ij} は画素と検出器の幾何学的配置で決まる。 C_{ij} が具体的にどのようなマトリクスになるかを見るために、画像を 16×16 画素、投影データを 16×16 投影としたときの C_{ij} の値を図4に画像として示す。この場合、 C_{ij} は 256×256 のマトリクスになり、横方向を画像 (j 番目の画素)、縦方向を投影データ (i 番目の検出器) として表示している。 C_{ij} を画像として見るとよくわかるが、ほとんどが値のない0となっている。ある角度において1つの画素から検出器に入る可能性があるのは最大で3つの検出器になる。ほかの検出器には値が入らないのでほとんどが0になる。

画像を 256×256 画素、投影データを 256×256 投影としたときの C_{ij} を作成し (この場合 C_{ij} は 65536×65536 マトリクスとなる)、(5) 式を実行した結果を図5に、(6) 式を実行した結果を図6に示す。それぞれ投影と逆投影が実行されている。

2. ML-EM 法

ML-EM 法は統計学的な理論に基づいて繰り返しの式 (逐次式) を算出している。その逐次式は以下のように表される。

$$\lambda_j^{(k+1)} = \frac{\lambda_j^{(k)} \sum_{i=0}^{N-1} \frac{y_i C_{ij}}{\sum_{j=0}^{M-1} C_{ij} \lambda_j^{(k)}}}{\sum_{i=0}^{N-1} C_{ij}} \quad (7)$$

ここで、 k は繰り返し回数を表す。 j は画像の画素番号を表し、すべての画素数は M である。 i は投影データの検出器番号を表し、すべての検出器の数は N である。 $\lambda_j^{(k)}$ と $\lambda_j^{(k+1)}$ はそれぞれ k 番目と $k+1$ 番目の画像の画素値に相当し、 y_i は実測した投影データである。 C_{ij} は検出確率である。この式を計算の手順に沿って分解して考えると以下ようになる。

① k 番目の画像から k 番目の投影を作成する。投影の式は (5) 式と同じである。

$$y_i^{(k)} = \sum_{j=0}^{M-1} C_{ij} \lambda_j^{(k)} \quad (8)$$

② k 番目の投影と実測した投影データとの比を求める。

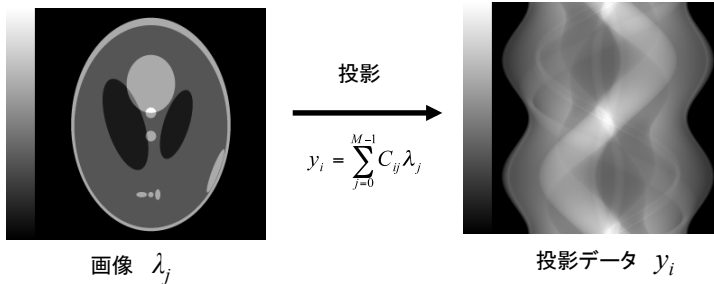


図5. C_{ij} を利用して投影を行った結果図

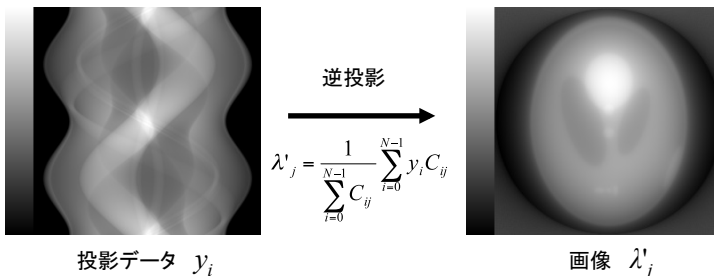


図6. C_{ij} を利用して逆投影を行った結果

$$y_i' = \frac{y_i}{y_i^{(k)}} \quad (9)$$

③ その比を逆投影する。逆投影の式は (6) 式と同じである。

$$\lambda_j' = \frac{1}{\sum_{i=0}^{N-1} C_{ij}} \sum_{i=0}^{N-1} y_i' C_{ij} \quad (10)$$

④ k 番目の画像に逆投影した画像を掛けて k+1 番目に画像を更新する。

$$\lambda_j^{(k+1)} = \lambda_j^{(k)} \cdot \lambda_j' \quad (11)$$

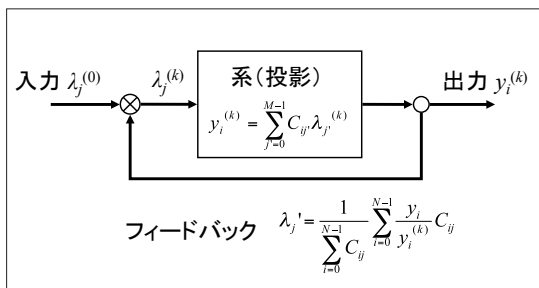


図 7. ML-EM 法をフィードバックの図で示したもののフィードバックの係数 λ_j' は比と逆投影で作成し、掛け算でフィードバックしている。

この手順をフィードバックの考え方で表すと図 7 のようになる。ただし、ここでは比と逆投影を使って係数 λ_j' を作成し、掛け算を使ってフィードバックを行っている。この更新の様子を実際の画像を入れて図 8 に示す。最初の入力 $\lambda_j^{(0)}$ はすべて値が 1 の画像である。

以上の計算手順を繰り返すことによって、画像 λ_j はその投影が実測した投影データに近づいていく。画像 λ_j の投影がほぼ実測した投影データに一致したところで、その画像 λ_j を再構成画像とする。繰り返しの打ち切りに関しては、経験的に行っているのが現状である。また初期値としては「正の値であること」という制限はあるが、一般には画像全体に一様分布を仮定するか、画像の内接円内の一様分布を仮定する。

Shepp ファントムの投影データから ML-EM 法で再構成した画像を図 9 に示す。繰り返しの回数が 1 回、5 回、10 回、50 回の画像を並べて表示している。繰り返しの回数を重ねるごとに原画像に近づいていく様子が見られる。また、再構成画像の評価値をプロットしたグラフを図 10 に示す。画像の評価には、元となる数値ファントムとそれぞれの再構成画像の各画素の差を絶対値にして平均をとったもの（平均

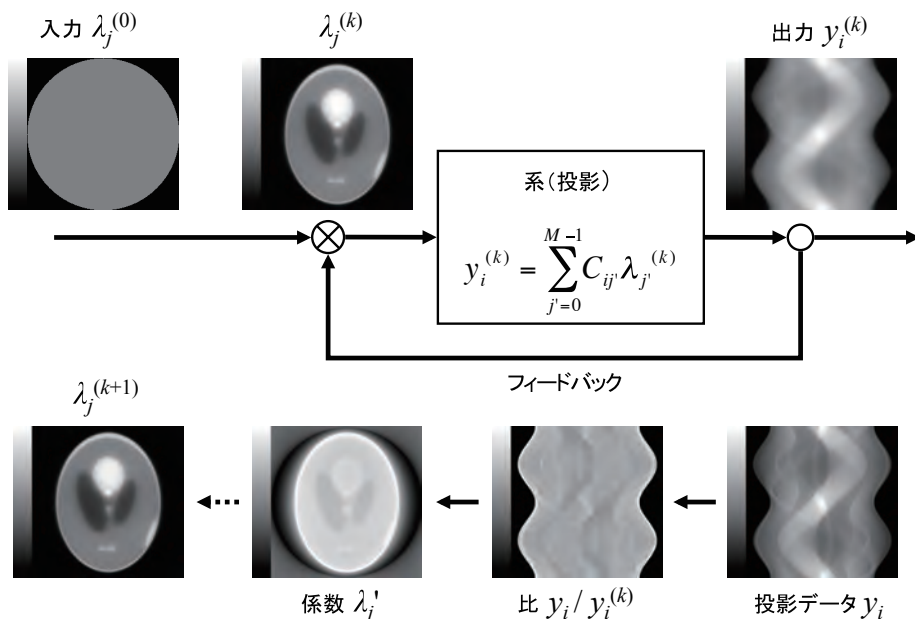


図 8. ML-EM 法をフィードバックの図に実際の画像を当てはめて示したもの

絶対誤差)を用いた。図10では、前回の繰り返しの方法との比較のため、ART法とSIRT法の評価値もプロットしている。ML-EM法は、出だしがSIRT法と同じぐらいから始まり、ART法やSIRT法よりも速く収束している。ART法とSIRT法では、差と緩和係数を使って更新用の係数を作成し、前の画像に加えることによって更新した。今回のML-EM法では、比を使って係数を作成し、前の画像に掛けることによって更新している。比と掛け算を使うことにより、安定した収束が見込める。

検出確率 C_{ij} に実際の測定系で起こり得る物理現象を組み込んでおけば、この影響を補正して画像再構成することができる。これがML-EM法の柔軟性を高くしている。実際にSPECTなどの画像再構成に、

- ① 画素 j から出た光子が検出器 i に到達するまでの減弱の割合 (減弱補正)
- ② コリメータによって画素 j から出た光子が距離に依存して広がりを持つことを考慮した割合 (深さに依存する分解能補正)
- ③ 被写体内で光子が散乱することによって広がりを考慮した割合 (散乱線補正)

を考慮した検出確率が使われている。

3. OS-EM 法

OS-EM法は投影データをいくつかの組(サブセット)に分割しておき、このサブセットに属するデータだけで、投影、逆投影、比較、更新を行い、それをサブセットごとに繰り返す方法である。すべてのサブセットでの更新を行った時点で、全体の更新1回分としている。サブセットを1としたときがML-EM法に相当する。また、サブセットの数を投影データの数に等しくしたときは、加減と乗除の違いはあるが、ART法の考え方と同じになる。通常サブセットは8や16などが使われる。投影数が16の場合のサブセットの分割例を図11に示す。

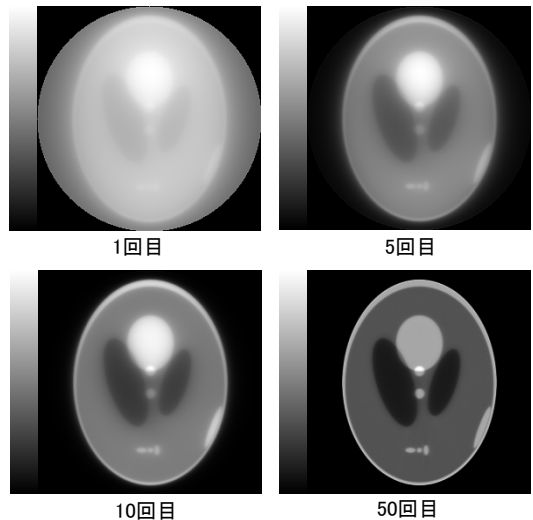


図9. ML-EM法の数値シミュレーション結果

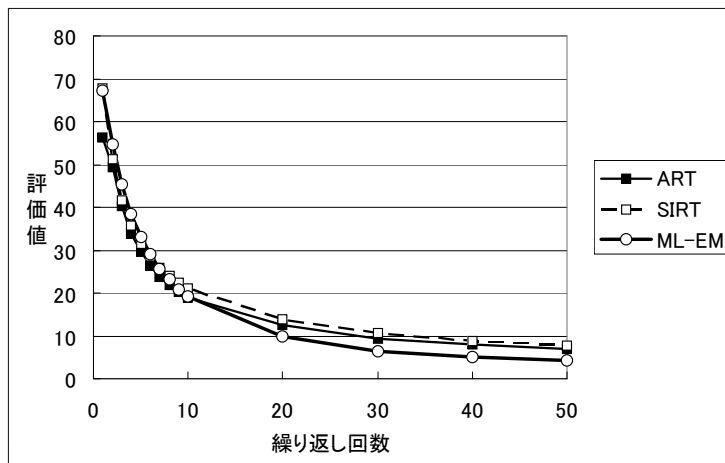


図10. ML-EM法およびART法とSIRT法の繰り返し回数に対する再構成画像の評価値

OS-EM 法の計算手順を以下に示す。

- ① 検出確率 C_{ij} を計算する。
- ② 初期画像 (内接円内を一様分布にするなど) を仮定する。
- ③ 初期画像をあるサブセットに属する角度に対してのみ投影を計算する。
- ④ 同じサブセットに属する投影データ y_i と、③で計算した投影との比を計算する。
- ⑤ ④で計算された比をサブセットに属する角度に対してのみ逆投影する。

- ⑥ 逆投影画像を初期画像 $\lambda_j^{(k)}$ に掛けてサブセットの更新画像 $\lambda_j^{(k,s=1)}$ を作成する。
- ⑦ サブセット更新画像を初期画像として③に戻り、次のサブセットに対して計算する。
- ⑧ すべてのサブセットの計算が終わったら、その更新画像を $\lambda_j^{(k+1)}$ とする。

OS-EM 法では、サブセットに分けることによって 1 回の繰り返しで画像を更新する回数が増え、結果として速く収束する。画像の更新回数 = (サブセット数) × (繰り返し回数) の関係が成り立ち、一

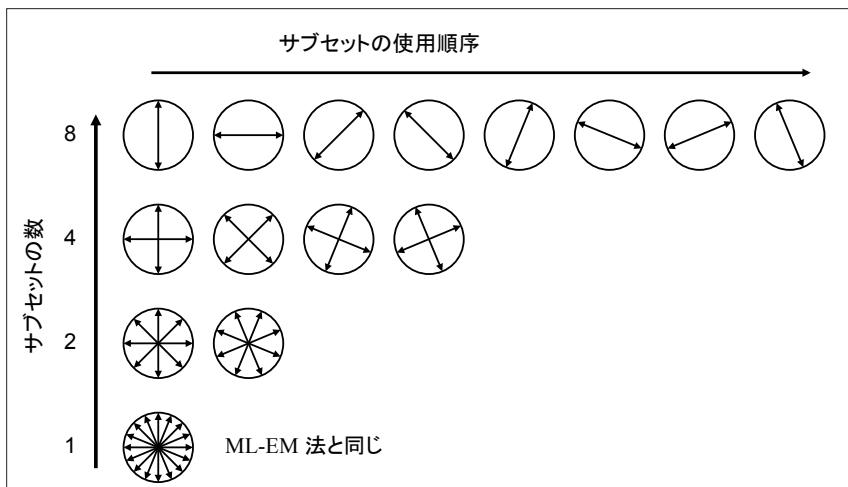


図 11. 投影数が 16 の場合のサブセットの分割例

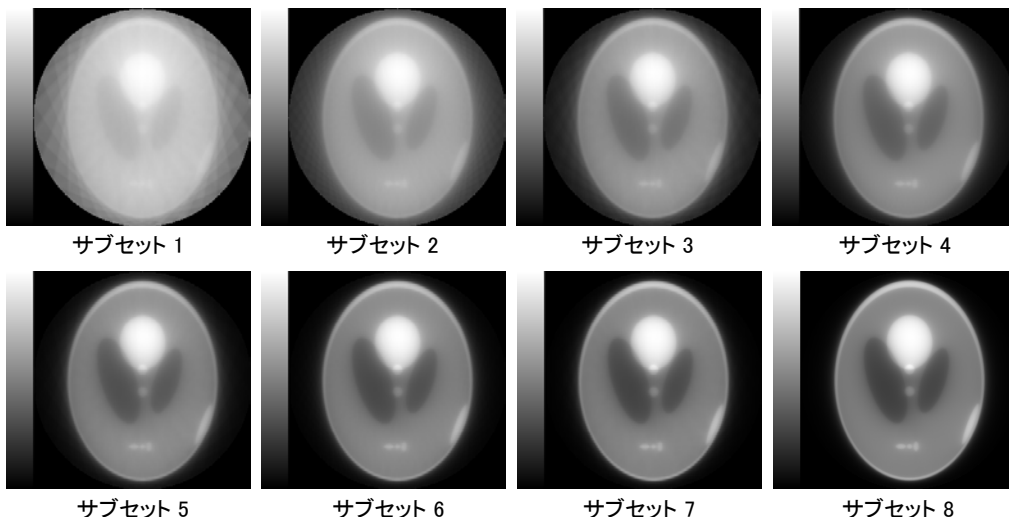


図 12. サブセットを 8 としたときのサブセットでの画像更新の様子

般にこの更新回数と同じであれば、ほぼ同様な再構成画像が得られる。サブセット数や使用する順序などは特に決まった規則はないが、なるべく離れた角度の投影データごとにサブセットを構成するのがよいといわれている。

サブセットを8としたときのサブセットでの8回の更新の様子を図12に示す。サブセット8の画像がOS-EM法での1回目の更新画像となる。サブセットを8としたときはOS-EM法の1回目の更新で実際には8回の画像更新を行っているの、ML-EM法に比べると更新速度が速くなるのがわかる。また、OS-EM法での繰り返し回数ごとの画像を図13に示す。画像の更新回数を括弧書きで示しているが、更新回数で見るとML-EM法の繰り返し回数の画像に近い画像となっている。

画像の評価値についてOS-EM法のサブセットでの更新回数とML-EM法の繰り返し回数を比較しているグラフを図14に示す。両者の評価値の変化は、ほぼ一致している。OS-EM法においては、繰り返し回数はサブセット8の場合、8個ごとになるので収束の速度は8倍程度速くなる。OS-EM法の1回の繰り返し時間がML-EM法での1回の繰り返し時間と同程度であるので、計算時間においても8倍程度速くなる。

サブセットを32にした場合のシミュレーション結果を図15に示す。収束の速度はさらに速くなって

いる。サブセットを256にした場合のシミュレーション結果を図16に示す。投影数を256にした今回のシミュレーションでは、サブセット256が最大になり、1投影ごとに画像を更新することになる。この場合、雑音を加えていないシミュレーションでも画像にざらつきが発生してしまう。実際のデータには雑音が含まれるので、サブセットは適当な大きさに抑えた方がよい。雑音などの変動成分をうまく押さえながらサブセットを最大にして、1回の繰り返し(画像更新は投影数だけ行う)で画像を作成する方法も提案されている。

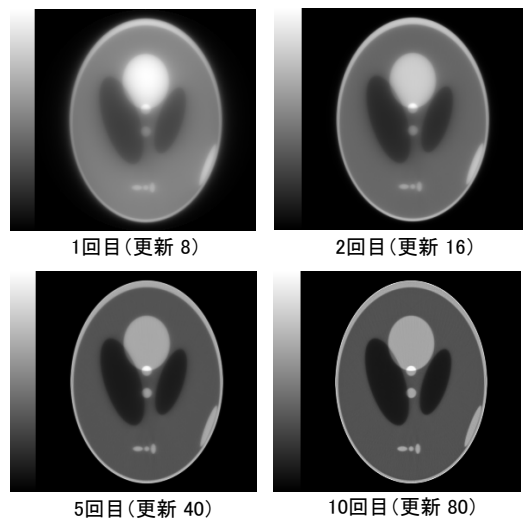


図13. OS-EM法(サブセット8)の数値シミュレーション結果

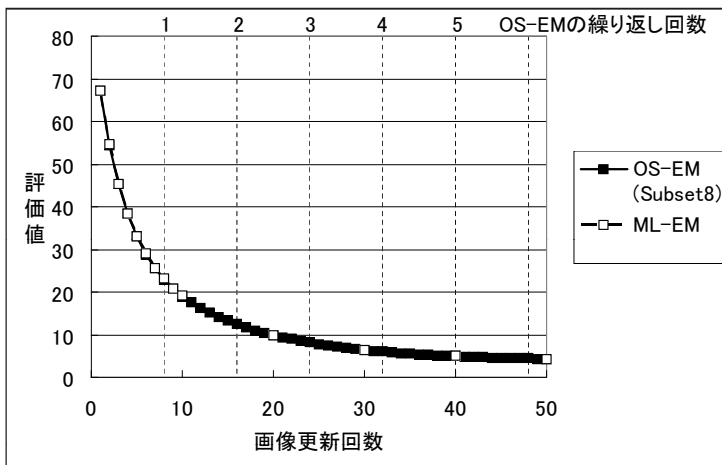


図14. OS-EM法とML-EM法の画像更新回数とOS-EM法の繰り返し回数での評価値

謝辞：本研究で使用したプログラムの開発は平成 17 年度～平成 23 年度首都大学東京共同研究費（富士フィルム RI ファーマ株式会社）、および平成 22 年度首都大学東京傾斜的配分研究費によるものである。

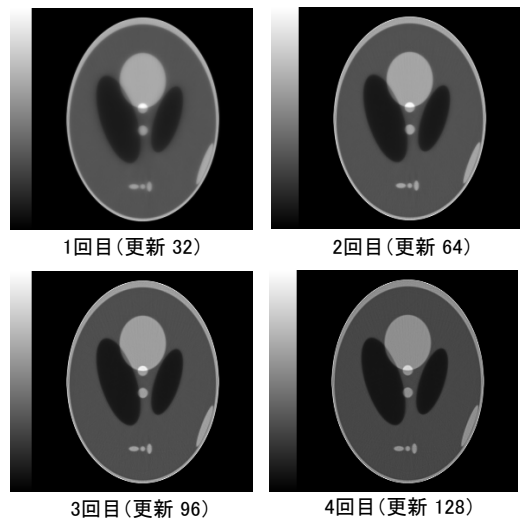


図 15. OS-EM 法(サブセット32)の数値シミュレーション結果

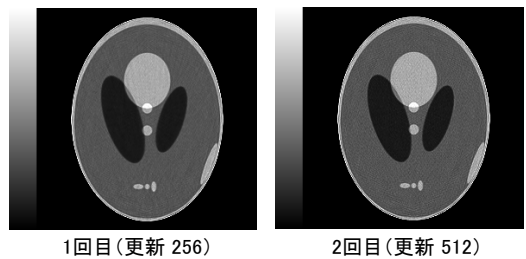


図 16. OS-EM 法(サブセット 256)の数値シミュレーション結果

サブセットが 256 の場合は、1 投影ごとに画像更新をするので ART 法の考え方と同じになる。