

SP- μ CT 画像の中野的再構成法

地質調査総合センター・中野 司 (tsukasa.nakano@aist.go.jp)

目次：

- (0) SP- μ CT 画像の中野的再構成法とは
- (1) プログラムのインストール
- (2) SP- μ CT 画像の再構成法
- (3) プログラムのリファレンスマニュアル
- (4) SP- μ CT 画像の再構成の実例

(0) SP- μ CT 画像の中野的再構成法とは
作文中

(1) プログラムのインストール

(1-0) 書庫ファイルのダウンロードと展開

SPring-8 の上杉健太郎さんのサイトにアップロードしてある同じ内容の書庫ファイル

<http://www-bl20.spring8.or.jp/~sp8ct/tmp/sp8ct.zip>

<http://www-bl20.spring8.or.jp/~sp8ct/tmp/sp8ct.taz>

のいずれかをダウンロードし、適当なディレクトリに展開する。

(1-1) DOS 窓用の実行ファイルのインストール

展開したディレクトリ `sp8ct/exe` を実行パスに登録するか、その中のファイルすべてを実行パスに登録済みのディレクトリにコピーする。なお、これらの実行ファイルは MinGW (Minimalist GNU for Win32) の GNU-C compiler でコンパイルしたものである。

(1-2) Cygwin 用の実行ファイルのインストール

以下のいずれかを行えばよい：

DOS 窓用の実行ファイルを流用する場合

先に述べた DOS 窓用の実行ファイルのインストール法と同じ。なお、Cygwin は DOS 窓の環境を継承するので、ディレクトリ `sp8ct/exe` を DOS 窓の実行パスに登録した場合にはそれを再度 Cygwin の実行パスに登録しなくてもよい。

実行ファイルを Cygwin 上でコンパイルする場合

コマンド `make` を

`make install.bin`

と入力して実行することを除けば次に述べる UNIX (Linux) 用の実行ファイルのインストール法と同じ。

(1-3) UNIX (Linux) 用の実行ファイルのインストール

まず、展開したディレクトリ sp8ct/src に移動して

```
make install
```

と入力する。これにより中野が用意した実行ファイルすべてがディレクトリ sp8ct/bin にインストールされるハズである。その後、このディレクトリを実行パスに登録するか、その中のすべてのファイルを実行パスに登録済みの適当なディレクトリにコピーすればよい。

なお、後で紹介する SPring-8 の高速計算機クラスタ pr47 では、その 5 台の計算機ノードが共有しているディレクトリ /home/image/bin に中野の画像再構成法で使用する実行ファイルすべてをインストールしてある。ただし、その際に、書庫ファイルを展開したディレクトリ sp8ct/src の中にあった Makefile を修正して、プログラムを Intel-C compiler を使ってコンパイルした（無修正の Makefile は GNU-C compiler を使う）。そして、念のため、修正した Makefile を pr47 のディレクトリ /home/image/src/sp8ct に残しておいた。

(2) SP- μ CT 画像の再構成法

(2-0) 処理に必要な作業用のディスク領域

SPring-8 の上杉健太郎さんが SP- μ CT のユーザに配布している画像再構成プログラムを使う場合と同様で、ここで紹介する画像再構成の手法でも以下の 2 種類のファイルを格納するための大容量のディスク領域が必要である：

測定データのファイル

処理速度の低下を甘受するなら、これらはハードディスクではなく CD や DVD の上に格納されていてもかまわない。

CT 画像のファイル

後で例を示すように、CT 画像として画像上での CT 値と画素値の対応関係を変えた複数種類のものを作ることが多い。これらのうち再構成した CT 値の情報をできる限り保持した、スライス画像ごとに CT 値を正規化した 16 ビット画素値の画像ファイルに要するディスク容量は、測定データのファイルの量と概ね同じになるハズである。

さらに、これらに加えて、中野的手法では以下のファイルのための作業領域が必要である：

シノグラムのデータファイル

上杉プログラムを使う場合とは異なり、中野的手法では画像再構成の前にスライスごとのシノグラムのデータファイルを作成しておく必要がある（シノグラムの作成と画像再構成を別々のプログラムで行うようにしてある）。これらに要するディスク容量は測定データの総量の 4 倍弱になるが、それは一時的に必要なものである（CD や DVD での永久保存は不要）。シノグラムのデータファイルは画像再構成の処理の直前に作成し、それが終了した後に消去すればよい。

処理を高速化するためには、シノグラムのデータファイルを測定データや CT 画像用のものとは別の高速なアクセスが可能なハードディスクに格納する方がよい。特に pr47 で処理を行う場合は、2 台の大容量ハードディスクの上にあらかじめ用意されているディレクトリ /home/tmp/ や /work/tmp/ の下に自分専用のディレクトリを作り、そこにシノグラムのデータファイルを一時的に格納すればよい（ただし、これらは不要になりしだい消去する必要がある）。

なお、上杉プログラムのユーザの多くは再構成した CT 画像のファイルを測定データのものと同じディレクトリに格納しているようだが、中野はそれに我慢できない。後で示すように、このような異なった種類のデータファイルは個別のディレクトリに分類して格納すべきである。通常、中野は CT の測定ごとにその測定番号を名前とするディレクトリを作り、その中に測定データや

CT 画像用のディレクトリを作成するなどして画像再構成の処理を行っている。

(2-1) 測定データファイルのチェック

SP- μ CT 画像の再構成には以下の測定データファイルが必要である：

output.log

測定した画像データの情報が書き込まれたテキストファイル。

dark.img もしくは d.img

暗電流強度 (dark) の画像のファイル。dark.img は上杉プログラム (mkdark.bat) を使って CT 測定の現場で複数の実測した dark の画像 (d01.img や d02.img など) から作ったハズのファイルである。この処理を忘れたために dark.img がない場合は以下のように入力すればよい (以下では実測した dark の画像が d0[0,1].img の 2 個だけの場合を示したが、もっと多数のものをパラメータとして書き並べてもかまわない)。

```
hp2mean d01.img d02.img d.img
```

これによって実測した dark の画像を平均した、dark.img の代用として使用可能な画像ファイル d.img を作成できる (中野的手法では d.img と dark.img の存在する方を使って画像再構成する)。

q*.img

入射 X 線強度 (I_0) もしくは透過 X 線強度 (I) の測定画像のファイル。それぞれのファイルが I_0 と I のどちらの画像であるかは前記の output.log に記述されている。

必要ならこれらのデータファイルを適当なディレクトリ (中野的には raw という名前のディレクトリ) にコピーする。そして、以下ではこのディレクトリ名を RAW とする。

以下のように入力すれば画像再構成に必要な測定データファイルの有無をチェックできる (不足しているファイルがあると警告を表示する)：

```
dit_tbl RAW d.tbl i.tbl t.tbl
```

ただし、dit_tbl は本来は hp2sg など (後で実行する測定データからシノグラムを作成するためのプログラム) の実行に必要なパラメータのファイル (d.tbl、i.tbl および t.tbl) を作るためのプログラムなので、測定データのディレクトリ名に続けてそれらのファイル名を指定した。ついにながら、このようなチェックを複数の測定データに対して行いたい場合は以下のように入力すればよい (嵐のように警告が表示される可能性がある)：

DOS 窓でチェックする場合

```
for /d %i in (RAWS/*) do @ dit_tbl RAWS/%i >nul
```

UNIX や Cygwin で B-shell (sh or bash) を使っている場合

```
for i in RAWS/*; do dit_tbl $i >/dev/null; done
```

UNIX や Cygwin で C-shell (csh or tcsh) を使っている場合

```
foreach i (RAWS/*)
  dit_tbl $i >/dev/null
end
```

ただし、ここでは測定データファイルが格納されている複数のディレクトリだけがディレクトリ RAWS の中にあると仮定している。また、先に説明した場合とは異なり、hp2sg 用のパラメータファイルの名前を指定しなければ dit_tbl は処理結果を標準出力に書き出す。ここでは測定データのチェックを行うだけなので、それを「ゴミ箱」に捨てた。

(2-2) 透過率もしくは投影の画像の作成と観察

CT 測定した視野の縁辺部に物体像が存在しないことがある。この部分の画像を再構成するのは時間の無駄なので、まずは測定した画像上の物体像を観察して、CT 画像を再構成するスライスの範囲などを調べるべきである。

測定した透過 X 線強度 (I) 画像は測定に使ったプログラム HiPic で観察できる。また、中野が書いたプログラムで I の画像 (例えば q002.img とする) を

```
hp2tiff q002.img q002.tif
```

のようにして TIFF に変換した後に適当な viewer で眺めてもよい。

しかしながら、入射 X 線強度 (I0) の空間分布の影響で I の画像では物体像を観察しづらい。そこで、dark、I0 および I の画像の同じ位置の画素ごとに以下の式で計算した (I0 の影響を取り除いた) X 線の透過率や投影値の画像を用いて物体像の広がりを調べると効率的である：

$$X \text{ 線透過率} = (I - \text{dark}) / (I0 - \text{dark})$$

$$X \text{ 線投影値} = \log(I0 - \text{dark}) - \log(I - \text{dark})$$

(2-2-0) 投影もしくは透過率の画像の作成

X 線の透過率や投影の画像を作成するプログラムは名前が

透過率画像の作成用：txy_tif (および hp3txy)

投影の画像の作成用：pxy_tif (および hp3pxy)

となっているだけで使用法はまったく同じなので、以下では投影画像の作成法だけを説明する。

まず、以下の入力によってディレクトリ RAW 中のファイルのうちで回転角が (単位は度；普通は 0 ~ 180 の範囲の浮動小数点数として取り扱われる値) の状態のサンプルを透視した投影画像の計算に必要なもののパス名を調べることができる：

```
[a] pxy_tif RAW
```

この結果、dark (d.img)、I0 (i.img) および I の測定画像 (t.img) のファイル名が以下のように表示されるはずである：

```
d.img = RAW/dark.img
```

```
i.img = RAW/qなんとか.img
```

```
t.img = RAW/qかんとか.img at angle =
```

ここで、`かんとか` は一番近い実測した回転角であり、また、t.img の測定時刻に一番近い時刻に測定された I0 の画像が i.img として選ばれている。

これらを用いて計算した投影画像の TIFF ファイルを以下のいずれかの入力で作成できる：

```
[b] pxy_tif RAW BPS TIFF
```

```
[c] pxy_tif RAW base step BPS TIFF
```

ここで、BPS は画像のビット数で 16 以下の自然数でなければならない。ただし、通常の画像 viewer はビット数が 1、2、4、8 および 16 以外の画像を取り扱えないものが多いので、BPS に 8 か 16 を指定するのが無難である。また、TIFF は投影画像の TIFF のファイル名である。中野的には投影画像専用のディレクトリ xp を作成し、その中に `= 0` と 180 の投影画像のファイル 000.tif と 180.tif を格納する (`= 90` の画像のファイル 090.tif を作成することもある)。

上記の [c] の base と step は投影画像上の各画素の投影値と画素値の対応関係を表す (浮動小数点数として取り扱われる) 値で、その関係は

$$\text{投影値} = \text{base} + \text{step} \times \text{画素値}$$

である。[b] のようにこれらを指定しないと、画像に出現した投影値すべてがちょうどおさまる base と step の値が選ばれる。また、その画像をファイルに格納した後に画像上に出現した投影

値の最小値と最大値が表示されるので、必要ならそれらの値を参考にして [c] の base と step の値を決めればよい。

(2-2-1) 投影もしくは透過率の画像の観察

以上のようにして作成した透過率もしくは投影の画像をじっくりと観察して以下の3組の座標値の範囲を（この優先順位で）決める：

物体像の影が特に濃い部分の上下方向の座標値の範囲

この座標値の範囲 ($z1 \sim z2$ とする) は後述するサンプルの回転軸の位置の直線近似 (サンプルの回転軸の傾きの推定) に用いる。画像の上下端まで物体像が見える場合でも画質のよい部分の範囲だけを選択した方がよい。なお、物体像の影が濃い部分は透過率の画像では暗く、投影画像では明るく表示されているはずである。

画像再構成するスライス (上下方向の座標値) の範囲

測定画像の上端付近には物体像がなく空気だけの場合も多い。このような部分のスライスを再構成するのは時間の無駄なので、物体像がある上下方向の座標値の範囲を調べておくとうい。

回転した物体像すべてがおさまる左右方向の座標値の範囲

画像再構成の計算量は測定画像 (透過率や投影の画像) の横画素数の自乗に比例する。このため、サンプルの回転軸から遠い測定画像の左右端の「余白」を取り除いたデータを使って画像再構成すれば、その計算量は激減する。このような測定画像の左右方向のトリミングは本来は測定の時に行うべきことだが、ここでの観察によって決めた画像の左右方向の座標値の範囲を指定すればシノグラムを作る時に行うこともできる。ただし、この範囲は回転した物体像のすべてがおさまるものでなければならない (つまり、 $0 \sim 180$ 度の範囲の様々な回転角の画像を観察してこの範囲を決める必要がある)。

これらの3組の座標値の範囲のうち後の2組は無理に決める必要はない。後述する処理でこれらの範囲を指定しないと計算時間が長くなるだけである。これに対して、 $z1 \sim z2$ は再構成画像の画質に影響するので、真剣に決める必要がある (それが必要になるサンプルの回転軸の位置の推定の処理と前後して決めるのがよいかもしれない)。

(2-3) サンプルの回転軸の位置の推定

画像再構成に必要な測定画像上でのサンプルの回転軸の位置 (画像上の横方向の座標値) を上杉プログラムではセンター値と呼んでいる。これに対して、中野的手法ではセンター値の符号を負に変えた値に相当する、DO (検出器の配置のオフセット; Detector Offset) と呼んでいる値でその位置を指定する：

センター値

測定画像の左端 (正確には、画像の左端にある画素の中央の点) からサンプルの回転軸までの距離。通常は画素の横幅 (検出器の間隔; dr) を単位とした値を使う。

DO (検出器の配置のオフセット; Detector Offset)

サンプルの回転軸の位置を原点とする測定画像上の横方向の座標系において画像の左端の画素 (の中央の点) を指す座標値。ただし、実寸の DO の値を使うこともあるので、DO の値が画素の横幅を単位としたセンター値の符号を負に変えた値であることを明示する場合は $r0/dr$ と書くことにする (この場合の $r0$ が実寸の DO の値である)。

特に問題がない測定データを得た場合、すなわち、

[0] CT 測定中にサンプルが動いた。

[1] サンプルの回転軸が大きく傾いていた。

- [2] 回転半径が大きくて物体像が検出器の視野からはみ出していた。
- [3] CT 測定に用いた X 線のエネルギーや露光時間の設定が不適切だったため測定した画像の S/N (Signal-Noise ratio) が非常に低かった。

のいずれでもない測定データを得た場合に、中野は DO の値の決定に (上杉プログラムのユーザのほとんどが行っている) センター値を試行錯誤で変えた換画像再構成を行わない。ここで紹介する手法でサンプルの回転角が 180 度だけ離れた方向から撮影した 2 枚投影画像 (これらを 0 度と 180 度の投影画像と呼ぶ) を用いて DO の値を推定し、念のためにその妥当性を画像再構成のテストで確認した後に、その値を本番の画像再構成の処理でそのまま使うことが普通である。

(2-3-0) サンプルの回転軸の位置の推定法の原理

サンプルの回転軸の位置の中野的推定法の原理を簡単に説明しておく。これは 2 段階の処理になっている。まず、0 度と 180 度のそれぞれの投影画像の上の同じ縦方向の位置にある横一列の画素が保持している投影値 $p_{000}(n)$ と $p_{180}(n)$ について考える。ここで n はそれぞれの画素の画像上での横方向の位置を示す整数値 (インデックス) で、 N を画像の横画素数とすると $n = 0 \sim N-1$ である。これらの投影値列はサンプルを「表」と「裏」から透視したモノなので、回転軸に関して対称になっているはずである。すなわち、 C を投影値列の上の回転軸の位置 (センター値) とすると、0 度の投影値の列上で回転軸から距離 d だけ離れた位置 $n = C+d$ の画素における投影値は 180 度の列上の位置 $C-d = C-(n-C) = C \times 2 - n$ の値と等しいはずである：

$$p_{000}(n) = p_{000}(C+d) = p_{180}(C-d) = p_{180}(C \times 2 - n)$$

実測した投影値にはノイズが混入しているため、上のような等号の関係式は実現しないが、それでも、測定データのパターンマッチングによって任意の (可能な) 位置 n に対して

$$p_{000}(n) \approx p_{180}(N-1-n)$$

と見なせる値 を推定できたなら、 C は以下の式で求まるはずである：

$$C = (N-1) / 2$$

中野的手法では 0 度と 180 度の投影値列の RMSD (投影値の差の自乗の平均の平方根 ; Root Mean Square of Differences) の計算によって測定データのパターンマッチングを行っている：

$$\text{RMSD}(n) = \sqrt{\sum_{n=0}^{N-1} \{ p_{000}(n) - p_{180}(N-1-n) \}^2 / (N-1)}$$

ただし、 $\sum_{n=0}^{N-1}$ は可能な n に関する $\{ \}$ 内の和の計算を意味する。

様々な n の値を指定して計算したもののうち $\text{RMSD}(n)$ が最小になる状態で 2 個の列の投影値のパターンが一致しているものと見なす。そして、その場合の n の値を先の式に代入して求めた C (および、DO の値 $r_0/dr = -C$) をここでの推定値とする。なお、RMSD が最小になる n の値は 1 (単位は画素の横幅) 刻みで探索できるので、 C や r_0/dr の値は原理的には 0.5 の精度で推定できるはずである。

このようにして測定した画像の縦方向の位置 (すなわち、CT 画像のスライス) ごとに DO の値を推定できる。しかしながら、投影値列ごとに S/N が異なるため、これらの推定精度は同じではない。投影画像上のノイズが一様に分布していると仮定すると (たいていはそうなっているようである)、投影値が相対的に大きい (物体像の影が濃い) 列ほど DO の値の推定精度は高いはずである。ここでは、それぞれの列の投影値の大きさを 0 度と 180 度の値の相互相関 (Cross Correlation ; CC) で測ることにする：

$$\text{CC}(n) = \sum_{n=0}^{N-1} \{ p_{000}(n) \times p_{180}(N-1-n) \}$$

ただし、この n は $\text{RMSD}(n)$ が最小になっている場合の値である。

さて、DO の推定値はスライスごとに少しずつシフトしていることが普通である。これは、物理的には、サンプルの回転軸が傾いているために生じたと考えられる (後述)。そこで、サンプルの回転軸の位置の推定の第 2 段階として、スライスごとに推定した DO の値をその位置 (スライ

ス番号 z) に関する以下のような 1 次式 (直線) で最小自乗近似する :

$$DO(z) = z \times \quad +$$

ただし、係数値 と はスライスごとの DO の推定値を使った最小自乗法で決める。

そして、画像再構成の処理では、それぞれのスライスに対して推定した値ではなく、スライス番号 z に応じたこの直線の上の DO の値を使用する。なお、この最小自乗近似では前記の CC の値をスライスごとの DO の値の「重み」としている。このため、CC が 0 に近い値になっている物体像がないスライスの DO の推定値 (これはデタラメな値のハズ) の寄与は抑えられているハズだが、それでも、これらが最小自乗近似で「悪さ」をすることがある。それを避けるためには、以前に透過率や投影の画像を観察して決めた物体像の影が濃いスライスの値だけを最小自乗近似で使うようにすればよい。

(2-3-1) サンプルの回転軸の傾き

スライスごとに推定した DO の値の直線近似によってサンプルの回転軸の傾きを求めることができる。しかし、これは実際の回転軸の傾きのうちの投影画像に平行な成分だけである (投影画像からそれに直交する X 線光路の方向の傾きを推定することは原理的に不可能)。また、回転軸の傾きが大きいと投影画像の横一列の画素ごとの DO の値の推定は不可能である。具体的には、サンプルの回転半径を R (投影画像の画素の縦幅を単位とする) として、絶対値が $1/R$ 以上の回転軸の傾きはここで紹介した方法で推定できない (逆に、そのような傾きになる場合は、ここでの推定結果を信用しない方がよい)。

サンプルの回転軸の傾きの投影画像の面と平行な成分を精密に求めるためには、ここで紹介したものとは別の手法を用いる必要がある。それを行った上で測定データを補正 (回転) するプログラム (irac シリーズ) も用意してあるが、ここではその説明は省略する。

(2-3-2) サンプルの回転軸の位置の推定プログラムの実行

前記の原理に基づいたサンプルの回転軸の位置の推定を行うためには、測定データファイルが入っているディレクトリを RAW として以下のいずれかを入力すればよい。

[a] DO_tbl RAW >DO.tbl

[b] DO_tbl RAW z1 z2 >DO.tbl

これらのいずれの方法で起動した場合も

D000 = RAW/dark.img

I000 = RAW/q0001.img

T000 = RAW/q0002.img at angle = 000.1200

D180 = RAW/dark.img

I180 = RAW/q1502.img

T180 = RAW/q1504.img at angle = 180.1200

のように表示される実測した dark の画像 (D???)、I0 の画像 (I???) および I の画像 (T???) から計算した 0 度と 180 度の投影画像を用いてスライスごとの DO の値を推定する。その後、[a] の起動法ではすべてのスライスの、また、[b] では $z = z1 \sim z2$ の範囲のスライスの DO の推定値の直線近似が行われる。これらの結果は標準出力に書き出され、上のようにしてテキストファイル DO.tbl にリダイレクトして保存すればよい。このファイルの各行には以下の値がタブコード区切りで書き込まれている :

1 行目

N 測定画像の横画素数 or CT 画像の横および縦画素数。

Z 測定画像の縦画素数 or CT 画像のスライス数。

2 ~ Z+1 行目

z CT 画像のスライス番号 (z = 0 ~ Z-1)。

DO そのスライスに対して推定した DO (r0/dr) の値。

RMSD

そのスライスの位置にある 0 度と 180 度の投影値列の RMSD の最小値 (それらの投影値列のデータに含まれるノイズの強度)。

CC そのスライスの位置にある 2 個の投影値列を DO の値に相当する配置にした場合の相互相関の値 (それらの列の各画素の投影値の自乗の総和 ; 投影値のパワー)。

Z+2 行目 (最後の行)

、 および

スライスごとに推定した DO の値を近似した直線

$$DO(z) = z \times \quad + \quad \pm$$

の係数値。ただし、 はサンプルの回転軸の傾き、 はその切片の値 (番号 0 のスライスにおける回転軸の位置)、そして、 は直線近似の推定誤差を表している。

z1 と z2

起動時に指定した DO の推定値の直線近似に用いたスライスの範囲。ただし、起動時の指定がなければこれらを書き込まない。

(2-3-3) サンプルの回転軸の位置の推定値の検討

上記の DO.tbl に書き込んだデータのうち最後の行の係数値 と が特に重要である。つまり、

先にも書いたようにここで使用している方法ではあまりに大きな回転軸の傾きを推定できない。サンプルの回転半径を測定画像の横幅 N の半分程度だと考えると、 の絶対値が $2/N$ を越えている場合にはここで行った直線近似の結果に問題がある。

それぞれのスライスに対して DO の値 (r0/dr) は 0.5 の精度で推定できているはずなので、この の値も 0.5 以下 (実用的には 1 以下) になっている必要がある。そうでない場合には投影値が低い (物体像の影が薄い) スライスの DO の推定値が「悪さ」をしている可能性がある。

や の値が大きい場合は透過率や投影画像を観察し直したり、 DO.tbl の 2 ~ Z+1 行目の値を検討したりして DO の推定値の直線近似に使うスライスの範囲 z1 ~ z2 を決め直し、先の [b] の起動法で DO.tbl を再度作成することが望ましい (この作業をおろそかにすると、高画質の再構成画像を得ることはできない)。

どのようにしても や の値が大きい場合は測定したデータが不良 (測定中にサンプルが動いた、など) である可能性が高い。その場合には (上杉プログラムを使う場合と同様に) 後述する画像再構成のテストで試行錯誤的に DO の値を決定せざるをえない。ここで推定したスライスごとの DO の値はその場合にも少しは参考になるはずである。

(2-4) シノグラムの作成

中野的手法では画像再構成に使うシノグラムのデータを測定データからあらかじめ作成しておく必要がある。先にも書いたように、これは画像再構成の処理の間だけ必要な一時的なデータであり、多量 (測定データの総量の 4 倍弱) のディスク容量を要する。そして、処理時間を短縮するためには、高速に動作するハードディスクの上に格納することが望ましい。以下ではシノグラムのデータファイルを一時的に格納するディレクトリの名前を SG とし、それを適当なハードディスクの上に作成済みだとする。

ディレクトリ RAW 中の測定データファイルを以前に

```
dit_tbl RAW d.tbl i.tbl t.tbl
```

と入力してチェックした場合は、以下のいずれかの入力でシノグラムのデータファイルを作成することができる：

```
[a] hp2sg d.tbl i.tbl t.tbl SG
[b] hp2sg d.tbl i.tbl t.tbl z1 z2 SG
[c] hp2sg d.tbl i.tbl t.tbl n1 z1 n2 z2 SG
```

また、以下のいずれかの入力を行えば、ファイル [d,i,t].tbl を作ることなくシノグラムのデータファイルを作成することができる：

```
[d] dit_tbl RAW | hp2sg SG
[e] dit_tbl RAW | hp2sg z1 z2 SG
[f] dit_tbl RAW | hp2sg n1 z1 n2 z2 SG
```

上記の [a] や [d] の入力を行うと、測定データが許す最大の画素数の CT 画像を再構成するためのシノグラムのデータファイルを作成できる。すなわち、測定した画像の横と縦の画素数をそれぞれ N と Z とし、スライス面内の横・縦画素数がともに N でスライス数が Z (スライス番号が $0 \sim Z-1$) の CT 画像用のシノグラムになる。これに対して、[b]、[c]、[e] および [f] では、スライス番号の値域が $z1 \sim z2$ の画像の再構成に必要なシノグラムになる。さらに、[c] や [f] のように指定すると、測定画像上の横方向の座標値が $n1 \sim n2$ の範囲の画素のデータだけを使って計算した、横と縦の画素数がともに $n2 - n1 + 1$ のスライス画像用のシノグラムになる。ただし、このような横方向にトリミングした測定画像から作ったシノグラムの再構成処理にはトリミングなしの画像を使って推定した DO の値 (ファイル DO.tbl に記されている値) をそのまま使用できない。この場合は推定値に $n1$ を加えた値を DO の値として指定する必要がある。

プログラム hp2sg は [a]、[b] もしくは [c] で指定されたテキストファイル d.tbl、i.tbl および t.tbl のそれぞれから dark、I0 および I の測定画像のファイルのパス名とそれらの測定時刻のデータを読み出してシノグラムを作成する。また、[d]、[e] もしくは [f] のようにして起動した場合には hp2sg はそれらを標準入力から一括して読み出す。その際に、t.tbl に記述されている I の画像の個数は CT 測定のサンプルの回転のステップ数 (投影数 M) と見なされる。さらに、それぞれの I の画像が t.tbl の上での並びの順番 ($m = 0 \sim M - 1$) に応じた回転角 ($180 \times m / M$ 度) の方向からサンプルを撮影したものであるとして hp2sg はシノグラムを計算する。

投影画像の場合と同様で、シノグラム上のそれぞれの画素の投影値の計算は dark、I0 および I の測定画像上の対応する画素の値を使って以下の式で行われる：

$$\text{投影値} = \log(I0 - \text{dark}) - \log(I - \text{dark})$$

この投影値は I と同じ回転角の方向からサンプルを撮影したものと見なされるが、その計算に I とは異なる時刻に測定した I0 や dark の値を使うわけにはいかない。そこで、I0 や dark の測定画像の枚数が 1 でないなら、hp2sg は測定時刻に関する線形補間で I の画像の測定時刻における I0 や dark の画像の値を推定し、投影値の計算に用いている。

こうして作成したシノグラムのデータはスライスごとに、ディレクトリ SG 中のスライス番号を名前としたファイルにそれぞれ格納される。これらは独自形式のバイナリファイルで portable ではない。つまり、シノグラムのデータファイルを異なった architecture の計算機に転送しても正常に取り扱えないことがある (例えば、Intel CPU を搭載した計算機で作ったモノはそのままでは HP の PA-RISC CPU の計算機で処理できない)。ここではシノグラムのデータファイルの形式に関するこれ以上詳しい説明は省略する。

シノグラムの作成にはかなりの処理時間を要する (データの量やハードディスクの性能にもよるが、数ギガバイトの測定データなら数十分もの処理時間を要するかもしれない)。それゆえ、次に説明する画像再構成のテストに用いる少数のスライスのシノグラムは本番の画像再構成用の多

数のモノとは別に作成する方がよいかもしれない。

(2-5) 画像再構成のテスト

前述のファイル DO.tbl の最後の行の の値が概ね 1 よりも小さければ、DO の推定値 (DO.tbl の最後の行の と の値) を使った本番の画像再構成に問題はないと思われるが、中野はその確認のために少数のスライスに対する画像再構成のテストを行うことにしている。なお、このことについてはこれ以上言及しないが、どのようにしても小さな の値が得られなかった (つまり、DO の値を推定できなかった) 場合に、以下で説明するものと概ね同じ手順で本番の画像再構成用の DO の値を試行錯誤的に決定することができる。

(2-5-0) 画像再構成のテストの実行法

画像再構成のテストのために、まず、投影画像などを観察して物体像の影が濃い部分のスライスを選択する (通常、中野はできるだけ離れた位置の 2 枚のスライスに対してテストを行う)。これらのスライス番号を z とすると、先の推定結果からその画像再構成に適した $r0/dr$ の値 $R0$ は、

$$R0 = z \times \quad +$$

である。この値の周囲の値を DO の値として指定して画像再構成を行う。つまり、 $R0$ の値がほぼ中央に来るように等間隔 (ここではその間隔を $step$ とする) に並んでいる適当な個数 ($count$ 個とする) の一連の値 $R1(order)$ (ただし、 $order = 0 \sim count-1$) を $r0/dr$ の値とする :

$$R1(order) = base + step \times order$$

$$\text{ここで、} base \quad R0 - step \times (count - 1) / 2$$

通常、中野は値 $step$ を 1、 $count$ を 5 程度とし、また、 $base$ の値を調節して整数になるようにした値 $R1(order)$ を以下の処理に使っている。

再構成のテストで得られる (スライスごとの) CT 画像を格納するディレクトリの名前を TEST (中野的には、このディレクトリ名は test/スライス番号)、また、それらの情報を書き込むテキストファイルの名前を TEST.log (中野的には、すべてのスライスに関する情報を test.log にまとめて書き込む) とする。以下のように入力すれば $R1$ のそれぞれの値を DO ($r0/dr$) の値とした複数個の画像の再構成処理を順次行うことができる。

```
mkdir TEST
```

```
sg2tgs_nai SG/z dr base step count t0 BPS TEST >TEST.log
```

ただし、上で指定しているパラメータの意味は以下の通りである :

SG/z

シノグラムのデータファイル名。なお、前述の方法で作成したシノグラムのファイル名には、すべてのスライスのモノが同じ文字数になるように、スライス番号 (z) の左に 0 を付加した文字列になっているモノもある。

dr 測定画像の画素の横幅 (検出器の間隔)。ただし、ここでは CT 値の値そのものを問題にしないので、適当な値 (1 など) を指定すればよい

base、step および count

再構成に使う $r0/dr$ を決める上記の $R1(order)$ のパラメータ。

t0 サンプルの回転角の初期値 (単位は度)。CT 画像上の物体像を回転させる角度を指定するが、今の場合は 0 とすればよい。

BPS

CT 画像の画素値のビット数。16 以下の自然数が許されているが、投影画像の場合と同様で 1、2、4、8 or 16 を指定すべきである。

TEST

再構成した複数の CT 画像を入れるディレクトリ名。それぞれの画像は再構成に用いた $r0/dr$ の符号を正にした値（つまり、センター値）を表す文字列（ただし、小数点は“_”に変える）を名前とする TIFF のファイルに格納される。

上記の入力で標準出力をリダイレクトしたテキストファイル TEST.log の各行には、再構成した CT 画像ごとの以下の 3 個の数値がタブコード区切りで並んでいる：

$r0/dr$

その画像の再構成に用いた DO ($r0/dr$) の値。

$f1$ と $f2$

その CT 画像に出現した CT 値の最小値と最大値。なお、これらの CT 値の単位は指定した dr の値の単位の数である。

(2-5-1) 再構成のテストで得た CT 画像の検討

このようにして再構成した CT 画像を適当な viewer で眺めるなどしてそれぞれの画質を比較・検討する。その際の注意点は以下の通りである：

- [0] ここで作成した CT 画像には、画像ごとに CT 値を最小値と最大値の範囲で正規化した画素値が入っている。それゆえ、画像の一部を詳しく調べるためには、適当な方法（例えば、ヒストグラム平滑化など）を使って CT 画像の表示輝度を強調する必要がある。
- [1] 物体像の上下に「影」や「ハロー」がある、もしくは、物体像の左右の端の輪郭線に「食い違い」が生じているなら、その画像の再構成に用いた DO の値は不適當である。
- [2] このような低画質の画像を見極める方が容易なので、 $R1$ （画像再構成に使った一連の DO の値）の両端の値を使った画像からはじめて、 $R1$ と画質が「 $R0$ （ここでの画像再構成に最適なハズの値）を対称中心とするような対応関係」になっているかどうかを調べるとよい。
- [3] CT 画像が最良の画質になっている DO の値を決めかねる場合は、先の TEST.log に記されている CT 値の最小値 $f1$ を調べるとよいかもかもしれない。データに含まれるノイズの影響で $f1$ は負の値になっているハズだが、経験的には、高画質の画像ほどその絶対値が小さい（0 に近い値になっている）。ただし、このことに理論的な根拠はなく、常にそれが成り立っているとは限らない。

DO の推定値の妥当性を以上のようにして確認できるハズである。これができなかった場合は測定データが不良だったと諦めるしかない。

(2-6) 本番の画像再構成

本番の画像再構成に先立って以下の 3 つのことを決めておく必要がある：

画像再構成処理に用いるプログラム

ここでは詳しい説明は省略するが、CT 測定でサンプルを 180 度回転しながら撮影した投影画像の枚数（サンプルの回転のステップ数もしくは投影数）を M すると、その回転軸からの距離が R （投影画像の画素の横幅を単位とする）の点（CT 画像上の画素）の CT 値を再構成するためには、理論的には $M \times R$ を満たしている必要がある。逆に、測定したデータ（シノグラム）をそのまま用いると、 $R > M/$ の点の CT 値を再構成するための数値計算（数値積分）が「不完全」になる。それを避けるためには、画像再構成の際にシノグラムを補間して投影数を「水増し」すればよい。

中野が書いた画像再構成プログラムにはこのようなシノグラムの補間を行う版と行なわない版の 2 種類のものがある。これらの違いは補間処理の有無と実行ファイルの名前だけで、使用法はまったく同じである。先の画像再構成のテストで用いた `sg2tgs_nai` のよ

うに、補間なしのプログラムの名前には "_nai" が付いている。そして、本番の画像再構成の処理でも sgs2tgs_nai と sgs2tgs の 2 個のプログラムを使用可能だが、以下の説明ではシノグラムを補間する sgs2tgs を使う場合だけを紹介する。物体像が回転軸の「近傍」にのみ分布している CT 画像を再構成する場合は処理時間（後述）が少なく済む sgs2tgs_nai を sgs2tgs の代わりに使用すればよい。なお、SPing-8 の上杉健太郎さんが SP- μ CT のユーザに配布している画像再構成のプログラムは中野の "_nai" 付きのものに相当する。

サンプルの回転軸の位置 (DO の値)

先に推定した DO の値を画像再構成のテストで確認できたなら、後で紹介する画像再構成プログラムの起動時にファイル DO.tbl の最後の行に記されている係数 と の値そのものを指定すればよい。ただし、それぞれのスライス画像はサンプルの回転軸がそのちょうど中央に来るように再構成されているので、回転軸の傾きを示す として 0 でない値を指定するとこれらを積み重ねた 3 次元 CT 画像は厳密には傾いたモノ (slant image) になる。それを避けるため、 の値として敢えて 0 を指定してもよい。また、試行錯誤的な画像再構成のテストによっておおよその DO の値だけしか決めることができなかつた場合も同様である。

CT 画像上での CT 値と画素値の対応関係

上杉プログラムを用いた画像再構成の処理ではスライスごとにそこに出現した最小値と最大値の範囲で CT 値を正規化した 16 ビット画素値の CT 画像をとりあえず作成し、その後、別のプログラム (tif_h2o) でこれらを変換して、すべてのスライスで CT 値と画素値の対応関係が揃っている画像 (通常の 3 次元画像処理プログラムが想定している画像) を作成するようである。

後で紹介する [a] のように入力すれば、中野的手法でもスライスごとに CT 値を正規化した画像を作成できる。しかしながら、測定した物質の LAC (X 線線吸収係数) や CT 値のおよその値があらかじめわかっている場合は、それを考慮して決めた係数値を画像再構成の際に指定して、すべてのスライスで同じ CT 値と画素値の対応関係にした CT 画像を作成するのが効率的である。つまり、後述する [b] の入力で、例えば、

$f_0 = 0$ (空気の LAC の値)

$df =$ サンプルの LAC の値 / 適当な画素値 (PV)

とした係数値 (8 ビット画素値の CT 画像を作る場合、中野は上の画素値 PV を 200 にすることが多い) を指定して、再構成したすべてのスライスの CT 値を単一の関係式

$$\text{CT 値} = f_0 + df \times \text{画素値}$$

or

$$\text{画素値} = (\text{CT 値} - f_0) / df$$

によって画素値に変換すれば、その後即座に 3 次元画像処理に使える CT 画像を得ることができる。また、出現する画素値の個数 (種類数) が減るので、この画像は CT 値を正規化した画像よりもずっと高い圧縮率でデータ圧縮してファイルに格納できるはずである (中野が書いたプログラムは画像データを LZW 法で圧縮して TIFF ファイルに書き込むようになっている)。

このように CT 値と画素値の対応関係を指定した画像再構成には利点が多いので、その実行を検討すべきである。なお、SP- μ CT で得たサンプルの CT 値 (LAC の推定値) は LAC の理論値の 0.8 ~ 0.9 倍の値 (この「装置係数」の値を と呼んでいる) になることがわかっているので、それを考慮して係数値 f_0 や df を決めるべきである。

(2-6-0) 本番の画像再構成の実行法

以下では本番の画像再構成で得たスライス画像を入れるディレクトリの名前を TG とする。ただし、TG は適当なハードディスクの上に作成済みだとする。中野的には、TG を以下のようなディレクトリ名にするのが普通である：

tg (以前は full もしくは slant だった)

これらはいずれもスライスごとに CT 値を正規化した 16 ビット画素値の画像を入れるディレクトリの名前である。以前は単一の DO の値を指定してすべてのスライスを再構成した場合には full (full value resolution image のつもり) に、また、係数値 と から計算した DO を使った場合は slant (slant image) にしていたが、現在はどちらの場合もディレクトリ名を tg にしている。

word もしくは byte

すべてのスライスで CT 値と画素値の対応関係を同じにした場合、DO の値の如何によらずこれらのディレクトリ名を使っている。ただし、画素値のビット数を 16 とした場合は word、それが 8 なら byte としている。

ディレクトリ SG の中のすべてのシノグラムのデータファイルに対して画像再構成の処理を行うためには、以下のいずれかを入力すればよい。

```
[a] sgs2tgs SG dr      t0 BPS TG >TG.log
[b] sgs2tgs SG dr      t0 f0 df BPS TG >TG.log
```

ただし、上で用いているパラメータの意味は以下の通りである：

SG シノグラムのデータファイルを入れたディレクトリの名前。ただし、これらのファイル名はそれぞれのシノグラムのスライス番号 (z) を表す数値でなければならない (先に説明した hp2sg で作成したものはそうなっている)。後でも説明するように、このスライス番号 z をもとにして計算した DO (r0/dr) の値を使ってそれぞれのスライス画像を再構成する。

dr 測定画像の画素の横幅 (検出器の間隔)。再構成した CT 値は dr に反比例 (1/dr に比例) した値になる。物質の LAC (X 線線吸収係数) やその推定値である CT 値は 1/cm を単位とするのが普通なので、dr として単位が cm の値を指定すべきである。

と

それぞれのスライス画像の再構成に使う r0/dr の値を計算するための係数値。スライス番号 z のスライスの再構成に使う DO の値は以下の値になる：

$$r0/dr = \quad + \quad \times z$$

なお、入力の際の値 と の指定の順番に注意せよ (先に作成した DO.tbl の最後の行に並んでいる順番と逆になっている)。

t0 サンプルの回転角の初期値 (単位は度)。正の値を指定するとスライス画像上の物体像はその角度だけ反時計回りに回転する。また、これが 0 なら、スライス画像を画面表示した場合の下方向が SP- μ CT の上流 (光源の方向) になる。

f0 と df

CT 値と画素値の対応関係を指定するための係数値。これらを指定すると CT 画像上のすべての画素で

$$CT \text{ 値} = f0 + df \times \text{画素値}$$

となる。また、これらの指定を省略すると、スライスごとに

$$f0 = \text{そのスライスに出現した CT 値の最小値}$$

$$df = (CT \text{ 値の最大値} - \text{最小値}) / \text{画素値の最大値}$$

が使われる (つまり、スライスごとにそこに出現した CT 値の範囲を正規化して決めた

画素値になる)。

BPS

CT 画像の画素値のビット数。16 以下の自然数を指定できるが、1、2、4、8 or 16 にしないと後の画像処理で困ることがある。

TG 再構成した複数のスライスの CT 画像を入れるディレクトリ名。それぞれのスライスの画像はスライス番号に ".tif" を付けた名前の TIFF ファイルに格納される。

上の入力で標準出力をリダイレクトしたテキストファイル TG.log の各行には、再構成したそれぞれのスライス画像に関する以下の 4 個の数値がタブコード区切りで列挙されている：

z スライス番号。

r0/dr

そのスライス画像の再構成に用いた DO (r0/dr) の値。

f1 と f2

そのスライス画像に出現した CT 値の最小値と最大値。

(2-6-1) 画像再構成に要する計算時間

1 枚のスライスの画像再構成に要する時間 T はその画像 (正方形) の各辺沿いの画素数を N、シノグラムの投影数を M として、概ね

$$T = N \times N \times M \times L \times$$

となる。ここで L はシノグラムの投影数の「水増し」の係数で、画像再構成に "_nai" が付いた名前のプログラムを使った場合は L = 1 である。そうでない場合は再構成に用いた DO の値にも依存するが、概ね以下の値になる：

$$L = \{ (N/2) \times \quad / M \} \text{ を切り上げた自然数}$$

また、は使用した計算機 (CPU) やプログラムの処理系 (compiler) などに依存する時定数である。中野が普段使っている環境では は以下の値であった：

Intel Itanium2 CPU (1.6 GHz) + Intel-C compiler (ver. 9.1)

4 ナノ秒

Intel Xeon CPU (2.4 GHz) + GNU-C compiler (ver. 3.2.2)

10 ナノ秒

なお、Xeon は Intel Pentium4 と似た architecture の CPU なので、Pentium4 を使った場合には

$$10 \text{ ナノ秒} \times 2.4 / \text{Pentium4 のクロック数 (単位は GHz)}$$

となるはずである。

(2-7) pr47 を用いた画像再構成の並列処理

pr47 (pr47.spring8.or.jp) は SPring-8 の BL47XU の側室に鎮座している計算機クラスタで、外部接続用とは別系統のギガビットイーサネットで結ばれた以下の 7 台の計算機からなる：

計算ノード (gw47、node1、node2、node3 と node4)

これらはいずれも前に紹介した高速な Intel Itanium2 CPU を 2 個ずつ搭載している 5 台の Linux (RedHat AS 2.0) マシン。ただし、それぞれのノードのメインメモリの量は異なっており、gw47、node1 と node2 の 3 台は 4、node3 は 8、そして node4 は 16 ギガバイトのメモリを積んでいる。また、gw47 は pr47 の窓口の計算機 (pr47 に外部からネットワーク接続すると gw47 に log-in することになる；これ以外のノードに外部から直に接続できない) であり、ソフトウェアの一部 (Intel compiler など) はその内蔵ディスクにだけインストールされている。

ファイルサーバ (fs47a と fs47b)

大容量 (2 テラバイト) のハードディスクをそれぞれ管理している Linux マシン。それぞれの計算ノードは fs47a の大容量ディスクをユーザのホームディレクトリ /home (正確には /misc/home) として、また、fs47b のディスクをユーザが自由に使用できる作業用のディレクトリ /work (/misc/work) として NFS (Network File System) でマウントして使うようになっている。

SP- μ CT のユーザは pr47 を利用して画像再構成処理などを行うことができるが、その許可や利用法などに関して管理者である SPring-8 の上杉健太郎さんに問い合わせる必要がある。

(2-7-0) LSF によるバッチ処理

通常の stand alone の計算機を使う場合と同様にして、ユーザは pr47 のそれぞれの計算ノードの上で自分のプログラムを実行できる (rlogin コマンドを使えば gw47 からそれ以外の計算ノードに log-in できる)。また、pr47 の 5 台のノードの合計 10 個の Itanium2 CPU を並列に稼働して処理を行うことも可能である。このような並列処理を実現する方法には複数個のものがあるが、ここでは LSF というソフトウェアを利用した、いわゆるバッチ処理で並列に画像再構成を行う方法だけを紹介する。LSF では依頼されたジョブ (通常のコマンド行からの入力に相当する処理の記述) を一旦待ち行列 (queue) に入れ、クラスタ上の計算ノードの空いている CPU でそれらを順番に実行する。LSF は依頼された時点でのカレントディレクトリなどの環境を再現してジョブを実行してくれるので、通常のコマンド入力によって実行できた処理の記述なら LSF のジョブとしてほぼそのままのもの (後述) を依頼することができる。

以下の 4 個の LSF のコマンド (これらの詳細は man コマンドで表示できる) を使えば、画像再構成のバッチ処理を問題なく行えるはずである。

lsload

pr47 の 5 台の計算機の稼働状況を表示する。

bsub ジョブの記述

LSF にジョブを依頼する。ただし、shell による即時解釈を防ぐために、ジョブとして記述した文字列に含まれている ">" などの特殊文字 (meta character) を「エスケープ」する必要がある (後に示す入力例を参照のこと)。

bjobs

LSF のジョブの状態 (待ち行列で待機中、など) を調べる。

bkill ジョブ番号

LSF に依頼したジョブのうち指定したジョブ番号のものの実行を取りやめる。ただし、ジョブ番号は bsub が表示してくれる番号で、bjobs で調べることもできる。

LSF は bsub で依頼したジョブの終了を E-mail で知らせてくれるが、ジョブを実行したノードにしかそれを配達できない。このため、bjobs で監視する以外にジョブの終了を知る簡便かつ確実な方法はない。また、この E-mail を放置するわけにもいかないなので、それを適当なファイルに書き出すように bsub に指定すべきである。ここでは以下のように入力して、LSF からのすべての E-mails の内容をテキストファイル LSF.log に書き込むことにする (ただし、E-mails の内容を常に同じファイルに書き込むためにはその絶対パス名を指定する必要がある) :

```
bsub -o LSF.log ジョブの記述
```

また、C-shell (csh or tcsh) を用いているなら、この代わりに

```
alias BSUB 'bsub -o LSF.log \!*
```

のように入力して定義した「別名」BSUB をコマンド bsub の代わりに使用すればよい (C-shell と同様にして bash でも「別名」を定義できるはずだが、中野はその方法を知らない)。以下の説明ではこのような BSUB を使用する。

(2-7-1) 画像再構成のテストの並列処理

中野的に 2 枚のスライス (z1 と z2 とする) に対して画像再構成のテストを行う場合は、以下のように入力すればそれを並列に処理できる :

```
mkdir -p test/z1 test/z2
BSUB sgs2tgs_nai SG/z1 dr ... BPS test/z1 \>test/z1.log
BSUB sgs2tgs_nai SG/z2 dr ... BPS test/z2 \>test/z2.log
```

ただし、ここでは sgs2tgs_nai の起動パラメータの記述の一部 ("..." の部分) を省略した。また、上のように、標準出力に書き出されるデータをファイルにリダイレクトするための特殊記号 ">" を "\" で「エスケープ」しており、これだけが通常の方法で実行する場合と異なっている。

(2-7-2) 本番の画像再構成の並列処理

複数個の測定データに対する本番の画像再構成は、以下のように入力して個々の測定に対する処理を並列に行えばよい (言うまでもないことだが、プログラム sgs2tgs にはそれぞれの測定に適した起動パラメータを指定する必要がある) :

```
BSUB sgs2tgs SG dr ... BPS TG \>TG.log
```

ただし、ここでも sgs2tgs の起動パラメータの記述の一部を省略した。また、この場合も、通常の方法で実行する場合との違いは特殊記号 ">" を "\" で「エスケープ」することだけである。

LSF ではジョブの個数に比例してバッチ処理の準備に要する時間が増大するので、あまりに多数のジョブを同時に依頼することは避けるべきである。個々の測定の複数のスライスを並列に処理する場合もスライスごとではなく、ある程度まとまった枚数のスライスに対する処理をジョブ化する必要がある。そのためには、LSF にジョブとして依頼する際に、指定したディレクトリの中の一部のシノグラムのデータファイルだけを処理するようにプログラム sgs2tgs にオプションの指定をすればよい。すなわち、ディレクトリ SG 中の Z 個のシノグラムのデータファイルのうちでそこ (ディレクトリ SG) に登録された「順番」order (= 0 ~ Z - 1) が

$$\text{order} = \text{start} + \text{leap} \times \text{index}$$

となる整数値 index (= 0 ~) に対応するものだけを再構成するように、プログラム sgs2tgs に以下のようなオプション指定を行えばよい :

```
sgs2tgs SG start leap count dr ... BPS TG ...
```

ただし、

start (= 0 ~)

画像再構成を行う最初のシノグラムの SG 上での「順番」。

leap (= 1 ~)

画像再構成を行うシノグラムの SG 上での「順番」の間隔 (飛び)。

count (= 0 ~)

1 以上の値を指定するとそれは処理すべきシノグラムの総数を指し、前記の式で index の値はそれ未満の範囲 (0 ~ count - 1) に限定される。また、値 0 を指定すると index の値域に上限はなくなり、前記の式の order の値が 0 ~ Z - 1 の範囲に落ちるすべての index の値に対応するシノグラムを処理する。

例えば、ディレクトリ SG 中にある 1000 個のシノグラムすべてを pr47 の計算ノードの 10 個の CPU のすべてを使って並列に再構成する場合は以下のいずれかを行えばよい。

連続した「順番」の 100 個ずつのシノグラムを個々のジョブで処理する場合
以下の 10 行を入力すればよい :


```

BSUB sgs2tgs SG 000 1 100 dr ... BPS TG \>TG0.log
BSUB sgs2tgs SG 100 1 100 dr ... BPS TG \>TG1.log
...
BSUB sgs2tgs SG 900 1 100 dr ... BPS TG \>TG9.log

```

もしくは、C-shell を用いているなら、これを以下の 3 行の入力で済ませてもよい :

```

foreach i (0 1 2 3 4 5 6 7 8 9)
  BSUB sgs2tgs SG ${i}00 1 100 dr ... BPS TG \>TG${i}.log
end

```

10 個飛びの「順番」のシノグラムを個々のジョブで処理する場合

以下の 10 行を入力すればよい :

```

BSUB sgs2tgs SG 0 10 0 dr ... BPS TG \>TG0.log
BSUB sgs2tgs SG 1 10 0 dr ... BPS TG \>TG1.log
...
BSUB sgs2tgs SG 9 10 0 dr ... BPS TG \>TG9.log

```

もしくは、C-shell を用いているなら、これを以下の 3 行の入力で済ませてもよい :

```

foreach i (0 1 2 3 4 5 6 7 8 9)
  BSUB sgs2tgs SG $i 10 0 dr ... BPS TG \>TG${i}.log
end

```

以上の 2 通りの方法のいずれでも pr47 の 10 個の CPU を使った本番の画像再構成の並列処理が始まるはずである。その後、それぞれの CPU (ジョブ) で再構成されたすべてのスライス画像はいずれもディレクトリ TG に格納され、また、それらに関する情報 (それぞれのスライスの画像再構成で使われた DO の値、および、そこに出現した CT 値の最小値と最大値) がジョブごとのテキストファイル TG[0-9].log に書き込まれる。最終的には、これら 10 個のジョブがすべて終了した後に以下のように入力して、それぞれのジョブが出力した画像再構成の記録を 1 個のファイル (TG.log) にまとめるなどすればよい :

```

sort TG[0-9].log >TG.log
rm TG[0-9].log

```

(2-8) 再構成画像上の CT 値と画素値の対応関係

先にも説明したように再構成した CT 値は以下の式で画素値に変換されてスライスごとの TIFF の画像ファイルに格納されている :

$$\text{CT 値} = f_0 + df \times \text{画素値}$$

この式の係数値 f_0 と df は sgs2tgs などのプログラムに陽に指定した値であったり、また、スライス画像ごとにそこに出現した CT 値の最小値と最大値の範囲から決められた値であったりするが、いずれの場合もこれらに関する情報がそれぞれの TIFF のファイルに「コメント」(TIFF の用語で Image Description Tag) として埋め込まれている。プログラム tg2tg を使えば、それをもとにした CT 画像上の画素の値 (CT 値および画素値) の出現頻度分布 (ヒストグラム) の調査や、CT 値と画素値の対応関係を変えた新しい CT 画像の作成を行うことができる。

以下のいずれかの入力を行えばプログラム tg2tg が起動する :

```

[a] tg2tg srcDir nameFile >HG.tbl
[b] tg2tg srcDir nameFile f0 df BPS >HG.tbl
[c] tg2tg srcDir nameFile dstDir >HG.tbl
[e] tg2tg srcDir nameFile f0 df BPS dstDir >HG.tbl

```

ただし、上で用いているパラメータの意味は以下の通りである :

srcDir

もとの CT 画像のスライス画像がおさめられているディレクトリの名前。

nameFile

ディレクトリ srcDir 中の TIFF ファイルのうち処理を行うべきものの名前 (ディレクトリ名は不要) を書き並べたテキストファイルの名前を指定する。ただし、srcDir 中のすべてのファイルを処理したい場合は "-" を、また、1 個のファイルだけを処理したい場合には "-" に続けてそのファイル名を指定すればよい。

f0、df および BPS

f0 と BPS は新しい CT 画像のすべてのスライスに適用する CT 値と画素値の対応関係の式で使う係数値。また、BPS はその画素値のビット数で、本文で紹介した他のプログラムの場合と同様に 1、2、4、8 or 16 を指定すればよい。

dstDir

新しい CT 画像のスライス画像を格納するディレクトリの名前。ただし、処理の前に自分でこれを作成しておく必要がある。なお、CT 値と画素値の対応関係を変えた新しいスライス画像のそれぞれは、もとのものと同じ名前の TIFF ファイルに格納される。

ヒストグラムを調べるだけの場合は上の [a] もしくは [b] のように、また、新しい画像を作る場合には [c] もしくは [d] のようにプログラム tg2tg を起動すればよい。そして、[a] や [c] のように新しい画像に関するパラメータ (f0、df および BPS) を指定しなければ、BPS はもとのスライス画像のビット数のうちの最大値に、また、f0 と df はもとの画像に出現したすべての CT 値の最小値と最大値の範囲を新しい画像の画素値でカバーする値になる。

プログラム tg2tg は常に新しい画像のヒストグラムのデータを標準出力に書き出すので、上記のように適当な名前のテキストファイル (ここでは HG.tbl とした) にリダイレクトすればよい。このファイルの各行には以下の 4 個の数値がタブコード区切りで並んでいる (ただし、tg2tg は下記の PN が 0 の「空」のピンのデータを出力しない) :

PV ヒストグラムのピン番号 (新しい画像の画素値)。

ff と fc

そのピンの下端と上端の CT 値 (単位は f0 や df のものと同じ)。

PN そのピンに落ちる画素の総数。

(2-9) SP- μ CT の画像再構成に関するその他のプログラム

作文中

(3) プログラムのリファレンスマニュアル

(3-0) hp2comment と hp2wl

起動法

hp2comment {HP}

hp2wl {HP}

説明

hp2comment は HiPic で測定した ITEX 形式の画像ファイルに埋め込まれているコメントの文字列を、また、hp2wl はその画像の横および縦の画素数を調べて標準出力に書き出すプログラムである。なお、起動時のファイル名 HP の指定は省略可能で、その場合には標準入力から ITEX 形式の画像ファイルのバイナリデータを読む (ただし、Windows の DOS 窓版のプログラムは標準入力からバイナリデータを正常に読み込めないかもしれない)。

(3-1) hp2mean

作文中

(3-2) hp2tiff

作文中

(3-3) hp3txy と txy_tif、および、hp3pxy と pxy_tif

作文中

(3-4) hp2DO と DO_tbl

作文中

(3-5) hp2sg、hp2sg.OM と dit_tbl

作文中

(3-6) sg2tg と sg2tg_nai

作文中

(3-7) sg2tgs と sg2tgs_nai

作文中

(3-8) sgs2tgs と sgs2tgs_nai

作文中

(3-9) tg2tg

作文中

(4) SP- μ CT の画像再構成の実例**(4-0) 処理に使うファイル**

ここまでに説明してきた SP- μ CT の画像再構成の処理を実際に試せるように、先に紹介した pr47 のディレクトリ /home/image/demo/040711j の中に以下のファイルを入れておいた：

raw/* (ディレクトリ raw の下の合計 402 個のファイル)

SP- μ CT で実測した測定データのファイル (output.log と *.img)。

*.bat もしくは *.csh

raw/* を使って画像再構成の処理を行うバッチファイルもしくはスクリプト。ただし、*.csh は UNIX や Cygwin で走る C-shell スクリプトである。また、*.bat は Windows の DOS 窓用のバッチファイルだが、後に記すように UNIX や Cygwin の shell (C-shell 以外も可) のスクリプトとして使えるものもある (shell スクリプトにはバッチファイルのような名前の制約がないため)。

これらのファイルをディレクトリ 040711j ごと自分の計算機 (pr47 を用いる場合は "cp -r" コマンドなどで自分のディレクトリ) にコピーし、その中に移動して以下の処理を行えばよい。

ディレクトリ 040711j/raw 中の測定データファイルは、大阪大学の土山明さんのビームタイムにおいて 2004 年 7 月 11 日にビームライン BL20B2 の SP- μ CT を使って撮影したモノである。

この測定（測定番号 040711j）の概要は以下の通りだった：

測定サンプル

直径が 0.3 ~ 0.5 mm の多数のビーズ球（orange glass ball；シリカガラス + ）。

X 線エネルギー

25 keV

測定画像 1 枚あたりの露光時間

10 sec.

透過 X 線強度 (I) の画像の枚数

合計で 360 + 1 枚を撮影。これらのうち 360 枚は CT 用にサンプルを 0.5 度刻みで 180 度回転しつつ撮影したもので、残りの 1 枚はサンプルの回転軸の位置決めのための、CT 用の最初の画像と 180 度だけ異なる方向から撮影した画像。

入射 X 線強度 (I₀) の画像の枚数

合計で 37 + 1 枚を撮影。これらのうち 37 枚は CT 用の I の画像の 5 枚ごとに撮影したもので、画像再構成では測定時刻に関するこれらの画像の線形補間によってそれぞれの I の画像の測定時刻における I₀ の値を計算する。また、残りの 1 枚はサンプルの回転軸の位置決めを使う 180 度の方向から撮影した投影の画像を作成する際に用いる。

暗電流画像の枚数

測定の最初と最後に 1 枚ずつ撮影。画像再構成ではこれら 2 枚の平均の画像を使う。

測定（投影）画像の画素数

1000（横方向）× 720（縦）

測定（投影）画像の画素（正方形）の辺長

5.83 μm (= 0.000583 cm)

ここでは、このような測定データから以下のサイズの CT 画像を再構成する：

画素数

1000 × 1000（スライス面内）× 720（スライス数）

画素（立方体）の辺長

5.83 μm (= 0.000583 cm)

後で説明するように、ディレクトリ 040711j に移動した後にそこにあるスクリプトを以下の順番で次々と起動して行けば SP-μCT 画像の再構成の一連の処理を実行できるはずである：

Windows の DOS 窓で実行する場合

1-0.bat、1-1.bat、2.bat、3.bat、4.bat、5.bat、6.bat、8-0.bat and 8-1.bat

stand alone の UNIX (Linux) 機もしくは Cygwin で実行する場合

1-0.bat、1-1.csh、2.bat、3.csh、4.bat、5.csh、6.bat、8-0.bat and 8-1.bat

pr47 もしくは LSF を使用できる UNIX (Linux) 機で実行する場合

1-0.bat、1-1.csh、2.bat、3.csh、4.bat、7-0.csh、7-1.csh、8-0.bat and 8-1.bat

or

1-0.bat、1-1.csh、2.bat、3.csh、4.bat、7-0.csh、7-2-0.csh、7-2-1.csh、8-0.bat and 8-1.bat

(4-1) 測定データファイルのコピーとチェック

中野は通常、SP-μCT で得た測定データのファイルすべてを raw という名前のディレクトリにコピーした後に画像再構成の処理を始める。今回の測定 040711j の場合はそれらを入れたディレクトリ raw を用意済みだが、そこには CT 測定の直後に上杉プログラムで作った dark.img は含まれていない。そこで、まず、1-0.bat を起動して画像再構成用の dark 画像を作成する：

```
hp2mean raw/d01.img raw/d02.img raw/d.img
```

これによって以下の警告が表示されるが、気にする必要はない：

```
raw/d01.img : time data not found (warning).
```

```
raw/d02.img : time data not found (warning).
```

この後、念のために以下のように入力して、CT 画像の再構成に必要な測定データのファイルが揃っているかどうかをチェックする：

Windows の DOS 窓の場合 (1-1.bat)

```
dit_tbl raw >nul
```

UNIX (Cygwin) の場合 (1-1.csh)

```
dit_tbl raw >/dev/null
```

これらいずれの場合もファイルの欠落を知らせる警告は出力されないはずである。

(4-2) 投影画像の作成と観察

まず、2.bat を起動して 0 度と 180 度の投影画像を作成する：

```
mkdir xp
```

```
pxy_tif raw 000 8 xp/000.tif
```

```
pxy_tif raw 180 8 xp/180.tif
```

この結果、以下の出力とともに 2 枚の投影画像を得ることができる：

```
d.img = raw/d.img
```

```
i.img = raw/q001.img
```

```
t.img = raw/q002.img at angle = 000.5000
```

```
-0.077514 1.882011
```

```
d.img = raw/d.img
```

```
i.img = raw/q397.img
```

```
t.img = raw/q396.img at angle = 180.0000
```

```
-0.076140 1.917379
```

これらの投影画像を観察すると、スライス番号が 240 ~ 480 の部分において物体像 (サンプルのビーズの像) の影が特に濃いことがわかるはずである。

(4-3) サンプルの回転軸の位置の推定

ここでは、投影画像の観察によってわかった、投影画像の上で物体像の影が特に濃いスライスのデータだけを用いてサンプルの回転軸の位置を推定する (3.bat もしくは 3.csh)：

```
DO_tbl raw 240 480 >DO.tbl
```

この結果、以下の出力とともにサンプルの回転軸の位置 (DO の値) を推定できるはずである。

```
D000 = raw/d.img
```

```
I000 = raw/q001.img
```

```
T000 = raw/q002.img at angle = 000.5000
```

```
D180 = raw/d.img
```

```
I180 = raw/q398.img
```

```
T180 = raw/q399.img at angle = 180.5000
```

UNIX (Cygwin) なら以下の入力でファイル DO.tbl の最後の行だけを表示できる (3.csh)：

```
tail -1 DO.tbl
```

それは以下のようにになっているはずである：

```
-0.000000 -498.500000 0.000019 240 480
```

すなわち、この測定では回転軸の傾きはほぼ 0 で、DO の値はどのスライスでも概ね -498.5 となっているようである。また、物体像の影が濃い部分のデータだけを使ったので、スライスごとの DO の推定値のバラツキ (0.000019) も非常に小さい。

(4-4) シノグラムの作成

4.bat を起動すればすべてのスライスに対するシノグラムのデータファイルをディレクトリ sg の中に作成できる：

```
mkdir sg
dit_tbl raw | hp2sg sg
```

なお、これらの多量のシノグラムは画像再構成の処理が終わりしだい消去すればよい。

(4-5) 画像再構成のテスト

画像再構成のテストを番号 360 のスライスに対して行うことにする。その結果の複数の画像ファイルをディレクトリ test/360 に格納する：

Windows の DOS 窓の場合 (5.bat)

```
mkdir test\360
```

UNIX (Cygwin) の場合 (5.csh)

```
mkdir -p test/360
```

先に推定した値 -498.5 を中央におく 1 刻みの 5 個の DO の値 (-500.5 ~ -496.5) を用いた画像再構成のテストは以下の入力によって行える (5.bat もしくは 5.csh)：

```
sg2tgs_nai sg/360 0.000583 -496.5 -1 5 0 8 test/360 >test/360.log
```

ディレクトリ test/360 中の 5 枚の TIFF 画像を比較することなどにより、先の推定値を DO の値とした再構成画像が最も高画質になっていることを確認できるはずである。

(4-6) 本番の画像再構成

測定サンプルが単純な組成・組織のピーズなので、以前に行った本番の画像再構成では CT 値と画素値の対応関係を陽に指定した。しかし、ここでは参考のため、複雑な組成・組織のサンプルに対する場合と同じにスライスごとに CT 値を正規化した 16 bit 画素値の CT 画像を作成する。6.bat を起動すれば、このような本番の画像再構成を行うことができる：

```
mkdir tg
sgs2tgs sg 0.000583 -498.500000 0.000000 16 tg >tg.log
```

(4-7) pr47 を用いた画像再構成の並列処理

pr47 で LSF を利用して画像再構成のテストや本番の画像再構成のバッチ処理を行う場合には、まず、7-0.csh、7-1.csh および 7-2-0.csh の 2 行目で記述されているように、C-shell を使って LSF にジョブを依頼するコマンド bsub の別名を定義しておくとう便利である：

```
alias BSUB 'bsub -o LSF.log \!*
```

この BSUB を使ってジョブを依頼すると、LSF はそれが終了した時に報告の E-mail をカレントディレクトリのファイル LSF.log に追記してくれる。

(4-7-0) 画像再構成のテストの並列処理

7-0.csh を起動すると、スライス番号 240 と 480 の 2 枚のそれぞれに対する画像再構成のテストを並列に行うことができる :

```
mkdir -p test/240 test/480
BSUB sg2tgs_nai sg/240 0.000583 -496.5 -1 5 0 8 test/240 \>test/240.log
BSUB sg2tgs_nai sg/480 0.000583 -496.5 -1 5 0 8 test/480 \>test/480.log
```

LSF は以下のように表示された番号を割り振って、これらのジョブを遂行する :

```
Job <13021> is submitted to default queue <normal>.
Job <13022> is submitted to default queue <normal>.
```

これらの処理はおよそ 1 分で終了するハズである。

(4-7-1) 本番の画像再構成のバッチ処理

7-1.csh を起動すると、pr47 の空いている CPU で測定 040711j の本番の画像再構成を行える :

```
mkdir tg
BSUB sgs2tgs sg 0.000583 -498.500000 0.000000 0 16 tg \>tg.log
```

この場合も以下のように表示された番号のジョブとして LSF は依頼された処理を行う :

```
Job <13023> is submitted to default queue <normal>.
```

このバッチ処理はおよそ 1 時間半で終了するハズである。なお、上記の sgs2tgs の代わりにプログラム sgs2tgs_nai を使うと、本番の画像再構成のバッチ処理に要する時間は 20 分以下になる。

(4-7-2) 本番の画像再構成の並列処理

pr47 の 10 個の CPU のすべてを使った、測定 040711j の 720 枚のスライスの画像再構成の並列処理は以下のようにする。まず、7-2-0.csh を起動して、画像再構成の 10 個のジョブを依頼する :

```
mkdir tg
foreach i (0 1 2 3 4 5 6 7 8 9)
  BSUB sgs2tgs sg $i 10 0 0.000583 -498.500000 0.000000 0 16 tg \>tg${i}.log
end
```

この結果、以下のように 10 個のジョブ番号が表示されるハズである :

```
Job <13024> is submitted to default queue <normal>.
Job <13025> is submitted to default queue <normal>.
Job <13026> is submitted to default queue <normal>.
Job <13027> is submitted to default queue <normal>.
Job <13028> is submitted to default queue <normal>.
Job <13029> is submitted to default queue <normal>.
Job <13030> is submitted to default queue <normal>.
Job <13031> is submitted to default queue <normal>.
Job <13032> is submitted to default queue <normal>.
Job <13033> is submitted to default queue <normal>.
```

この後、bjobs コマンドを使ってそれぞれのジョブの進捗状況を監視する。処理時間は 10 分程度のハズである。そして、すべてのジョブが終了した後に 7-2-1.csh を起動して、それぞれのジョブの画像再構成の記録を 1 個のファイルにまとめるなどすればよい :

```
sort tg[0-9].log >tg.log
rm tg[0-9].log
```

(4-8) 再構成画像上の CT 値と画素値の対応関係

今回の CT 画像上のサンプル（ビーズ）はシリカガラスなので、それに近い組成の鉱物である石英を「標準物質」とする。すなわち、再構成した 16 bit 画素値の CT 画像を 8 bit 画素値のものに変換する際に石英の LAC の理論値を「単位」とした以下のような CT 値と画素値の線形な対応関係を使うと、人に説明する場合や後々の画像解析の際に便利である：

CT 値 0（空気の CT 値）に対する画素値を 0 とし、
石英の LAC の理論値に相当する CT 値を画素値 PV とする。

問題は、シリカガラスに含まれている重元素の影響で実際のビーズの CT 値がどの程度の値になっているか、ということである。以下では 16 bit 画素値の CT 画像のヒストグラムからそれを読みとり、8 bit 画像への変換の際の「単位」となる画素値 PV を決めることにする。

(4-8-0) 画素値および CT 値のヒストグラムの作成

8-0.bat を起動して再構成した 16 bit 画素値の画像のヒストグラムを調べてみる：

```
tg2tg tg - >tg_hg.tbl
```

この後、以下の 2 個の数値を含む行が表示されるが、これらはディレクトリ tg 中のすべてのスライス画像の CT 値の最小値と最大値である：

```
-4.374080 33.054323
```

tg_hg.tbl をグラフなどにして眺めると、CT 値が 5.3 ~ 5.4 (1/cm) の付近にサンプルのものと思われるヒストグラムのピークがあることがわかる。

(4-8-1) 画素値と CT 値の対応関係の変更

今回の CT 画像の測定で用いた 25 keV の X 線エネルギーにおける石英（組成は SiO₂、密度は 2.65 g/cc とした）の LAC の理論値は 3.662520 (1/cm) である。16 bit 画素値の CT 画像を変換した 8 bit の画像において先に調べたサンプルを代表する CT 値を 2 値化などの画像処理に都合の良い画素値（150 ~ 200 程度の値）に対応させるためには、石英の LAC の理論値に対応する画素値 PV を 125 程度にすればよい。つまり、新たに作成する 8 bit 画素値の CT 画像に適用する CT 値と画素値の対応関係を以下のようにする：

$$\text{CT 値} = \text{画素値} \times 3.662520 / 125 = \text{画素値} \times 0.02930016$$

8-1.bat を起動すれば、再構成した 16 bit 画素値の CT 画像（tg）からこの対応関係に従った 8 bit 画素値の CT 画像（byte）とそのヒストグラムのデータファイル（byte_hg.tbl）を作成できる：

```
mkdir byte
tg2tg tg - 0 0.02930016 8 byte >byte_hg.tbl
```