6mct/recipe.txt : Tue Jan 23 11:13:28 JST 2018

SIXM の画像再構成などの手順

[0] プログラムのインストール法
[1] 作業用ディレクトリと測定データの準備
[2] 測定データのチェックなど
[3] MF のための RP 画像の観察
[4] R[I,R] 画像の作成
[5] 作成した R[I,R] 画像に対するオプションの処理
[6] R[I,R] 画像上の物体像を囲む bounding box の選択
[7] スライスごとの値の「時間変動」や鉛直方向のサンプルの「動き」のチェック
[8] サンプル回転軸の位置の推定
[9] サンプル回転軸の位置を変えた画像再構成のテスト
[10] 吸収および位相 CT 画像の再構成とそれらの確認
[11] 吸収 CT 画像に対する後処理
[12] 位相 CT 画像に対する後処理
[13] 長さと表示輝度のスケールバーを付けた吸収と位相 CT 画像の作成（オマケ）

[0] プログラムのインストール法

書庫ファイルのダウンロードと展開

```
% cd ☒/Desktop
% wget http://www-bl20.spring8.or.jp/¥☒sp8ct/tmp/6mct.taz
% tar xzf 6mct.taz
% rm 6mct.taz
```

CPU 用のプログラムのコンパイル

```
% cd 6mct
% make
```

GPU 用のプログラムのインストール

```
% cd cuda
% vi Makefile
    NVCC =nvcc -O3 -arch=sm_50
    CUDA =/usr/local/cuda
% make
% mv *_g ..
% cd ..
```

ImageJ のランチャーとマクロのインストール

```
% cp -p ImageJ.csh ☒/Desktop/ImageJ
% cp -p tlrb.txt ☒/Desktop/ImageJ/macros
```

C-shell の設定ファイルへの追加

```
% vi ☒/.cshrc
    set path=(☒/Desktop/6mct $path)
    alias IJ '☒/Desktop/ImageJ/ImageJ.csh ¥!*'
```

[1] 作業用ディレクトリと測定データの準備

測定ごとのディレクトリを作成し、その中に移動してすべての作業を行う：

```
% mkdir 170516c
% cd 170516c
    /dev/shm/170516c
```

SIXM の測定で得た２個のファイルをリンクしておく（コピーでも良い）：

```
% ln -s /work1/tsukasa/1705/170516c/a.log .
% ln -s /work1/tsukasa/1705/170516c/a.HIS .
```

[2] 測定データのチェックなど

SIXM の測定データファイルのバイト数などを調べる：

```
% ls -trlL
    total 41275208
    -rwx------ 1 tsukasa tsukasa    14074443 May 16 18:52 a.log
    -rwx------ 1 tsukasa tsukasa 42251730944 May 16 18:52 a.HIS
```

ファイル a.log に記されている SIXM の測定のパラメータ値を調べておく：

```
% head -1 a.log
    551,601,100,1,0
```

これらのコンマ区切りの数値のうちの最初の３個の意味は以下の通りで、それらを後述するプログラム his2raw の起動パラメータとしてそのまま指定する：

```
scans = 551
    各 view の最初の scan はゴミなので、R[I,R] 画像の横画素数は 550
views = 601
    CT に使用する 180 度回転のステップ数は 600 views
darks = 100
    データファイル a.HIS の先頭に格納されている暗電流 RP 画像の個数
```

データファイル a.HIS に格納されている RP 画像のサイズなどを調べる：

```
% check_his a.HIS ¦ uniq -c
    42251730350
    331251 128      498     2
```

ただし、check_his などが出力した数値の意味は以下の通り：

```
42251730350 ≦  42251730944
    有効な RP 画像の総バイト数。
    この値はファイル a.HIS のバイト数以下のはず。
331251 == 551 (scans) * 601 (views) + 100 (darks)
    有効な RP 画像の個数。
    ファイル a.log に記されている値から計算した値と一致するはず。
128
    RP 画像の横画素数。
    正確には、16 個づつの binning 後の画素数。
498 = Nz
    RP 画像の縦画素数。
    この値と等しい枚数の CT 画像のスライス画像を再構成できる。
2
    RP 画像のタイプ（2 == 16 ビット整数画素値の画像）。
    後述する his2raw はこのタイプの RP 画像しか処理できない。
```

これら以外に以下の SIXM の以下のパラメータ値を知っておく必要がある：

```
Dz = 108.1 (nm)
    RP 画像の画素の鉛直方向の辺長。
    この値は以下の処理では陽に指定しないが、
    RP 画像から作成する R[I,R] 画像の画素の鉛直方向の辺長、
    および、それらから再構成する３次元 CT 画像のスライスの厚さとなる。
Dr = 100 (nm) = 100e-7 (cm)
    SIXM のスキャンのステップ幅。
```

この値は位相 CT 画像の再構成処理では陽に指定しないが、
それを含めた３次元 CT 画像のスライス画像の正方形画素の辺長となる。
SDD = 6220 (mm) = 6220e-3 (m)
サンプル回転軸と検出器面の間の距離
Dp = 6.5 (um) * 16 (binning) = 104 (um)
RP 画像の画素の水平方向の辺長。
RP 画像が正立像なので、Dp として負の符号を付けた -104 を指定する。
RA0 = 0 (degree)
サンプル回転角の初期値。
非０の値を指定すれば軸回転した画像を画質の低下なしで再構成できる。
他のパラメータ値と区別するため、以下では RA0 として +0 を指定する。

なお、後述する画像再構成の処理では Dr、SDD および Dp として再構成する値に
応じた上記の単位の値を指定する必要がある：

吸収 CT 画像
単位が 1/cm の LAC の値を推定するので cm 単位の Dr の値を指定する。
位相 CT 画像
単位が 1e-6 (100 万分の１) の RID の値を推定するので、
SDD の単位に対して 100 万分の１の単位の Dp の値を指定する。

[3] MF (mountain flattening；山の平坦化) のための RP 画像の観察

R[I,R] 画像の作成時に RP 画像に対して MF の処理を行うかどうかを判断する。
そのためには複数の RP 画像を吟味することが望ましいが、ここではとりあえず
R[I,R] 画像の「中央」の画素の値の計算に使う１個の RP 画像だけを調べる：

R[I,R] 画像の「中央」の画素の位置
view = views / 2 = 300
scan = scans / 2 = 275
layer = Nz (RP 画像の縦画素数) / 2 = 249

これらとファイル a.log に記されていた値を用いてファイル a.HIS から１個の
RP 画像を抽出し、その画像 (もとのRP 画像) とそれに対してパラメータの値
Wm (mountain width) = Ws (samplineg width) = 5 および 10
を指定して MF を行った結果の RP 画像などを ImageJ で表示してみる：

```
% rp+mf_ij.csh a 300 275 249 5,5 10,10
    -380.329987     63949.281250
    -380.329987     63949.281250
```

これらの画像や同時に表示されるラインプロファイルを観察して
もとの RP 画像の「山」の高さが 2^16 = 65536 に近い「飽和した値」か？
Wm と Ws の値ごとの MF の結果のプロファイルに「不自然さ」がないか？
を確認し、MF の処理の実行の是非やその処理で用いる Wm と Ws の値を決める。
なお、rp+mf_ij.csh は MF の処理を行うたびにもとの RP 画像が保持している
画素値の値域を上記のように (標準エラー出力に) 書き出す。これらのうちの
最大値から「山」の高さが「飽和した値」になっているか否かを判断できる。

[4] R[I,R] 画像の作成

新しい２個のディレクトリをまず作成し、それらの中に a.log の値を指定して
プログラム his2raw で抽出した a.HIS の上の RP 画像、もしくはそれに MF
を行った結果の RP 画像のいずれかから計算した R[I,R] 画像を書き込む：

```
% mkdir ri rr
% his2raw a.HIS 551 601 100       ri rr > raw.log   # MF を行わない場合
% his2raw a.HIS 551 601 100 5  5  ri rr > raw.log   # Wm = Ws = 5
% his2raw a.HIS 551 601 100 10 10 ri rr > raw.log   # Wm = Ws = 10
```

言うまでもないことだが、上記の３通りのいずれか、もしくは Wm と Ws に任意
の値を指定した起動法で his2raw を１度だけ実行すれば良い。なお、ここで例と
した測定 170516c ではもとの RP 画像上の「山」が「飽和した値」だったので、
Wm = Ws = 10 を指定した上記の３番目の行の入力によって MF を行った R[I,R]
画像を作成した。

[5] 作成した R[I,R] 画像に対するオプションの処理

必要なら以下のような R[I,R] 画像の画素値のビット数の削減処理を行っても良い。

```
float (32 ビット画素値) の R[I,R] 画像を 16 ビットのものに変換
    % t2t_float ri - 16 ri > /dev/null
    % t2t_float rr - 16 rr > /dev/null

float (32 ビット画素値) の R[I,R] 画像を８ビットのものに変換
    % t2t_float ri - 8 ri > /dev/null
    % t2t_float rr - 8 rr > /dev/null
```

これらの処理を行うと R[I,R] 画像の容量が激減するが、それを用いた画像再構成
の結果が float の画像を使ったものと完全に同じになるかどうかは定かではない。

[6] R[I,R] 画像上の物体像を囲む bounding box の選択

ImageJ のマクロ tlrb を使って R[I,R] 画像のいずれかを表示し、それらの上の
物体像を完全にカバーする bounding box のひろがりを調べる：

```
% IJ -macro tlrb rr 1 10 tlrb.log
```

ただし、ここでは物体像の境界の識別が容易な RR 画像 (ディレクトリ rr) を指
定した (RI 画像を使いたければディレクトリを ri に置き換えれば良い)。また、
処理時間短縮のため RR 画像のうち１番目から 10 枚おきの view だけを表示する
よう指定した (すべての view を表示したければ「1 10」を削除すれば良い)。

画像下部のスライダ使えば指定した複数の view の画像のそれぞれを表示できる。
マウス左ボタンをドラッグしてそれらの画像の上に bounding box の長方形を描く。
その後、マウスの右ボタンを押すと bounding box の情報を示すウィンドウが開く
ので、その下部に並んでいる以下の３つのボタンのいずれかをクリックする：

Yes
表示されている bounding box の情報を記録した後に処理を続行する。
No
表示されている bounding box の情報を記録せずに処理を続行する。
Cancel
マクロ tlrb による bounding box の選択の処理を終了する。

上記の Yes により bounding box の情報がファイル tlrb.log にも書き込まれる。
Cancel (および ImageJ の終了) を行った後にそれを確認すれば良い：

```
% tail -1 tlrb.log
    100     50      120     470
```

これら４個の数値は R[I,R] 画像の上の物体像に関する以下の情報を示しており、
後述するプログラム "r[i,r]*" のそれぞれに起動パラメータとして指定する：

top = 100
物体像の上部の空気の領域の幅、もしくは、物体像の上端の座標値
left = 50
物体像の左側の空気の領域の幅
right = 120
物体像の右側の空気の領域の幅
bottom = 470
物体像の下端の座標値

なお、以下のようにすれば選択した bounding box を確認することができる：

```
% IJ -macro tlrb ri 1 10 100 50 120 470
% IJ -macro tlrb rr 1 10 100 50 120 470
```

[7] スライスごとの値の「時間変動」や鉛直方向のサンプルの「動き」のチェック

プログラム r[i,r]2m[p,r] に top、left と right の値を指定して R[I,R] 画像から平均投影値（MP）と平均屈折量（MR）の画像を作成する：

```
% ri2mp ri - 100 50 120 mp.tif
    -0.019094       0.119755

% rr2mr rr - 100 50 120 mr.tif
    -1.014936       0.893499
```

M[P,R] 画像の横軸は view（左右端がそれぞれサンプル回転角０度と 180 度）に、また、縦軸は R[I,R] 画像の縦軸に対応している。そして、M[P,R] 画像の画素値は
    top、left と right の値が指す空気の領域の画素の値を用いて
    R[I,R] 画像のそれぞれの画素の値から計算した
    X 線投影値（XP）と X 線屈折量（XR）の画像
のそれぞれの上の横一列の画素の値の平均値である。これらの値が view に関して大きく変動している（厳密には一定値になっていない）と高画質の CT 画像を再構成することができないので、M[P,R] を眺めてその様相を調べる：

```
% xv mp.tif mr.tif
```

M[P,R] 画像の上の「横方向に伸びている一定値の縞模様」が傾斜している場合は SIXM の測定中にサンプル（もしくはサンプルが載っているステージ）が鉛直方向に動いた可能性が高い。view に関して定常的（steady vertical creep）だと仮定できるなら、その動きを画像再構成の際に補正できる。ただし、そのためには回転角が０度と 180 度の X[P,R] 画像の上のサンプル像の鉛直方向の変位量（縦ズレ）の値 Dz を指定する必要がある。

後述するプログラム r[i,r]2rc を使えば値 Dz を推定できる。また、オプションの起動パラメータ（/dev/null）の指定により r[i,r]2m[p,r] でも自身が作成した M[P,R] 画像から値 Dz を推定できる。下記の r[i,r]2m[p,r] のそれぞれの出力の２行目の最初の値（今の場合はどちらも -2）が Dz の推定値である：

```
% ri2mp ri - 100 50 120 mp.tif /dev/null
    -0.019094       0.119755
    -2      4.369533e-03

% rr2mr rr - 100 50 120 mr.tif /dev/null
    -1.014936       0.893499
    -2      3.253222e-02
```

[8] サンプル回転軸の位置の推定

サンプル回転角が０度と 180 度の X[P,R] 画像の位置関係を変えながらそれらに含まれているサンプル像を比較することにより、SIXM の測定時のサンプル回転軸の位置（いわゆる「センター値」）を推定する：

```
% ri2rc ri - 100 50 120 100 470 0.25 0.25 ri2rc.tif
    206     343     241
    -124    124     -3

% rr2rc rr - 100 50 120 100 470 0.25 0.25 rr2rc.tif
    206     343     241
    -124    124     -6
```

物体像の領域だけに限定すれば０度と 180 度の画像の比較の精度が向上するので、ここでは空気の領域の拡がりを指す top、left と right の値に続けて物体像が占めている縦方向の範囲を指す top と bottom の値をプログラム r[i,r]2rc の起動時に指定した。その後の２個の値「0.25 0.25」は２画像の位置関係を変えるために行う横と縦方向の並進移動のそれぞれの総量の範囲を指す、画像の横と縦の幅を１とした相対値である。

上記の r[i,r]2rc のそれぞれの２行の出力の最初の２個の値が指定した相対値に対応した回転軸の位置と縦ズレの調査範囲の実際の値域である。そして、それらの後の各行の最後の値が回転軸の位置（１行目；センター値）と SIXM の測定中に生じたサンプルの縦ズレの推定値（２行目；前述の Dz）である。

測定 170516c では X[P,R] 画像から推定した回転軸の位置はどちらも同じ値 241 であり、それを吸収と位相 CT の両方の画像再構成で指定すれば良いはずである。なお、後述する画像再構成のテストの結果にもよるが、原理的には吸収と位相 CT の画像再構成の際に指定するサンプル回転軸の位置は同じ値にすべきである。

測定 170516c では r[i,r]2rc のそれぞれで推定した２個の縦ズレ Dz は同じ値になっていない。さらに、それらは前述の r[i,r]2m[p,r] で推定した Dz の値とも異なっている。つまり、ここで例として処理した 170516c のサンプルは測定中に鉛直方向に少しだけ動いたようだが、その動きを steady vertical creep だと仮定して補正するのは難しい。

なお、r[i,r]2rc で推定した回転軸の位置や縦ズレの値が指定した範囲の両端の値に一致している場合、それらの推定値は信頼できない。その場合には物体像が占める縦方向の範囲を変えて r[i,r]2rc を再実行してみるべきである。また、内部構造が特異（例えば、内部が一様）な物体像に対して回転軸の位置や縦ズレの値の推定処理を正常に行えないこともある。その場合は r[i,r]2rc の起動時に指定したファイル r[i,r]2rc.tif に格納されている「RMSD のマップ」の画像を観察して対応策を考える必要があるが、ここではその詳細の説明は省略する：

```
% xv           ri2rc.tif rr2rc.tif   # RMSD のマップを白黒で表示する
% xv -preset 4 ri2rc.tif rr2rc.tif   # 擬似カラー表示（低い値が青色）
```

[9] サンプル回転軸の位置を変えた画像再構成のテスト

念のため、先に推定した値（241）の前後のサンプル回転軸の位置を指定した画像再構成のテストを行う。ここでは以下のパラメータを指定する：

```
再構成するスライスの位置 = ( top + bottom ) / 2 = 285
テストする回転軸の位置の最小値 = 241 -0.5 * 7 = 237.5
テストする回転軸の位置の間隔 = 0.5
テストする回転軸の位置の総数 = 1（推定値）+ 7*2（その前後）= 15
```

画像再構成のテストは CPU で走る r[i,r]_stg_t と GPU を使う r[i,r]_stg_g のどちらのプログラムで行っても良い（以下では両者の実行法を併記する）。ただし、CPU 用を使う場合は実行前に稼働するスレッド数の設定を行っておく必要がある：

```
% setenv CBP_THREADS 8      # ８個のスレッドで並列に画像再構成を行う
```

吸収と位相 CT の画像再構成のテスト用のディレクトリ test_[a,p] のそれぞれを作成しておく。ここまでで説明したパラメータの値を指定して r[i,r]_stg_[t,g] を実行すれば、それぞれのディレクトリの中に
    指定したスライス位置の１個の sinogram と
    回転軸の位置を変えた複数個の tomograms
の２種類の画像を作成できる：

```
% mkdir test_a
% ri_stg_t ri - 100 50 120 285 100e-7       237.5 0.5 15 +0 test_a > test_a.log
% ri_stg_g ri - 100 50 120 285 100e-7       237.5 0.5 15 +0 test_a > test_a.log

% mkdir test_p
% rr_stg_t rr - 100 50 120 285 6220e-3 -104 237.5 0.5 15 +0 test_p > test_p.log
% rr_stg_g rr - 100 50 120 285 6220e-3 -104 237.5 0.5 15 +0 test_p > test_p.log
```

なお、steady vertical creep の補正を行う場合は先に推定した縦ズレの値 Dz
をサンプル回転角の初期値（+0）の直前に指定すれば良い（ただし、ここで例と
した測定 170516c に対しては steady vertical creep を仮定できないので不要）：

```
% mkdir test_a
% ri_stg_t ri - 100 50 120 285 100e-7       237.5 0.5 15 Dz +0 test_a > test_a.log
% ri_stg_g ri - 100 50 120 285 100e-7       237.5 0.5 15 Dz +0 test_a > test_a.log

% mkdir test_p
% rr_stg_t rr - 100 50 120 285 6220e-3 -104 237.5 0.5 15 Dz +0 test_p > test_p.log
% rr_stg_g rr - 100 50 120 285 6220e-3 -104 237.5 0.5 15 Dz +0 test_p > test_p.log
```

その後、まずはファイル「test_[a,p]/スライス番号.tif」の sinogram の画像を
観察して測定中のサンプルの不穏な「動き」の有無を確認する：

```
xv test_a/285.tif test_p/285.tif    # sinograms of layer 285
```

ファイル「test_[a,p]/スライス番号_回転軸の位置.tif」が回転軸の位置を変えて
再構成した画像である。これらを個別に眺めて回転軸の位置の違いに伴う CT 画像
の画質を評価しても良いが、それよりも回転軸の位置が異なる画像を並べたものを
眺める方が効率的である。以下のようにすれば吸収と位相 CT のそれぞれのテスト
用のディレクトリの下にある再構成画像すべてを並べた画像を作成できる：

横方向優先の順番で５×３のマトリックス状に並べる場合

```
% pile_h.csh test_a 1 1 15 5 3 test_a.tif
% pile_h.csh test_p 1 1 15 5 3 test_p.tif
```

縦方向優先の順番で３×５のマトリックス状に並べる場合

```
% pile_v.csh test_a 1 1 15 3 5 test_a.tif
% pile_v.csh test_p 1 1 15 3 5 test_p.tif
```

これらの画像を詳細に観察して、偽像のない高画質な CT 画像の再構成に最適な
サンプル回転軸の位置を決定する：

```
xv test_a.tif test_p.tif
```

[10] 吸収および位相 CT 画像の再構成とそれらの確認

サンプル回転軸の位置を決定した後（ここでは吸収と位相 CT のどちらに対しても
同じ値 241.0 とした）、本番の画像再構成を行う。テスト用の場合と同様に CPU
用と GPU 用の合計２×３個のプログラムのいずれかを利用できる（以下ではそれら
の実行法を併記する）。ただし、CPU 用のプログラムを実行する場合には稼働する
スレッド数（８とする）を事前に設定しておく必要がある：

プログラム r[i,r]2tg を実行する場合

```
% setenv THREADS 8
```

プログラム r[i,r]_tg_t を実行する場合

```
% setenv CBP_THREADS 8
```

テストの場合と同様に吸収と位相 CT の再構成画像用のそれぞれのディレクトリを
作成した後、これまでに紹介したパラメータの値を指定して画像再構成プログラム
r[i,r]2tg もしくは r[i,r]_tg_[t,g] を実行する：

```
% mkdir tg_a
% ri2tg   ri - 100 50 120 100e-7       241.0 +0 tg_a > tg_a.log
% ri_tg_t ri - 100 50 120 100e-7       241.0 +0 tg_a > tg_a.log
% ri_tg_g ri - 100 50 120 100e-7       241.0 +0 tg_a > tg_a.log

% mkdir tg_p
% rr2tg   rr - 100 50 120 6220e-3 -104 241.0 +0 tg_p > tg_p.log
% rr_tg_t rr - 100 50 120 6220e-3 -104 241.0 +0 tg_p > tg_p.log
% rr_tg_g rr - 100 50 120 6220e-3 -104 241.0 +0 tg_p > tg_p.log
```

なお、steady vertical creep の補正を行う場合は先に推定した縦ズレの値 Dz
をサンプル回転角の初期値（+0）の直前に指定すれば良い（ただし、この補正は
プログラム r[i,r]2tg では実行できない；また、ここで例とした測定 170516c
に対しては steady vertical creep を仮定できないので、この補正は不要）：

```
% mkdir tg_a
% ri_tg_t ri - 100 50 120 100e-7       241.0 Dz +0 tg_a > tg_a.log
% ri_tg_g ri - 100 50 120 100e-7       241.0 Dz +0 tg_a > tg_a.log

% mkdir tg_p
% rr_tg_t rr - 100 50 120 6220e-3 -104 241.0 Dz +0 tg_p > tg_p.log
% rr_tg_g rr - 100 50 120 6220e-3 -104 241.0 Dz +0 tg_p > tg_p.log
```

再構成した画像をざっと見るため、テストのものと同様に吸収と位相 CT のそれ
ぞれの等間隔になるように選び出した 15 枚のスライスを５×３もしくは３×５の
マトリックス状に並べた画像を作成してみる。

```
スライスの間隔 = ( bottom - top + 1 ) / 15 ≒  24
最初のスライス = top + ( bottom - top + 1 - 24 * ( 15 - 1 )) / 2 ≒ 117
```

横方向優先の順番で５×３のマトリックス状に並べる場合

```
% pile_h.csh tg_a 117 24 15 5 3 tg_a.tif
% pile_h.csh tg_p 117 24 15 5 3 tg_p.tif
```

縦方向優先の順番で３×５のマトリックス状に並べる場合

```
% pile_v.csh tg_a 117 24 15 3 5 tg_a.tif
% pile_v.csh tg_p 117 24 15 3 5 tg_p.tif
```

これらの画像を眺めて画像再構成の結果に不備がないかを確認する：

```
% xv tg_a.tif tg_p.tif
```

[11] 吸収 CT 画像に対する後処理

再構成した浮動小数点数画素値の CT 画像を画像処理に適した整数画素値の画像
に変換する。そのためにまず、再構成した LAC の値（CT 値）の値域を調べる：

```
% mm.csh tg_a
    -375.003174     1699.017944
```

SIXM に限らずこれまでの X 線 CT の測定ではすべてのスライスで上記のような
CT 値の値域全体を 16 ビット画素値にマッピングした word 画像を作成していた。
その作成時に指定が必要な 16 ビット画素値１階調ごとの CT 値の増分の値

```
(1699.017944-(-375.003174))/(2^16-1) = 0.0316475336537728
```

も先と同じ C-shell script ”mm.csh” で計算できる：

```
% mm.csh tg_a 16
    -375.003174     1699.017944     0.0316475336537728
```

このようにして得た値を指定して吸収 CT 画像の word 画像とその画素値ヒスト
グラムのデータファイルを作成する：

```
% mkdir word_a
% t2t_f.csh tg_a   -375.003174 0.0316475336537728 16 word_a > word_a.txt
```

また、word 画像を作成せずにその画素値ヒストグラムだけを得ることもできる：

```
% t2t_float tg_a - -375.003174 0.0316475336537728 16 -      > word_a.txt
```

その後、再構成した CT 値の全域のヒストグラムを表示してサンプルの大部分を占める「主要物質」の CT 値を調べる：

```
% plot_x11.csh word_a.txt
```

その際に CT 値の範囲を限定したければ以下のように指定すれば良い：

```
% plot_x11.csh word_a.txt 0 500     # CT 値 = 0 ～ 500 の部分を表示
```

表示した画素値ヒストグラムからサンプルの主要物質に相当するピークの CT 値を読み取り、その 1/100（この値に理論的根拠なし）の値を後述する byte 画像の８ビット画素値１階調ごとの CT 値の増分とする。

測定 170516c の吸収 CT 画像のヒストグラムのピークの CT 値 ≒ 124 (1/cm)
→ byte 画像の８ビット画素値１階調ごとの CT 値の増分 = 1.24 (1/cm)

このようにして決めた CT 値の増分の値を指定して CT 画像の byte 画像とその画素値ヒストグラムのデータファイルを作成する。

```
% mkdir byte_a
% t2t_f.csh tg_a 0 1.24 8 byte_a > byte_a.txt
```

先と同様に byte 画像の等間隔 15 枚のスライス画像を５×３もしくは３×５のマトリックス状に並べた画像を作成してみる：

横方向優先の順番で５×３のマトリックス状に並べる場合

```
% pile_h.csh byte_a 117 24 15 5 3 byte_a.tif
```

縦方向優先の順番で３×５のマトリックス状に並べる場合

```
% pile_v.csh byte_a 117 24 15 3 5 byte_a.tif
```

また、byte 画像の画素値ヒストグラムの PostScript の図も作成する：

```
% plot_a.csh ./byte_a.txt > byte_a.ps
```

なお、byte 画像の画素値０と 255 には「ゴミ」が混じっているので、
画素値０に対応する CT 値の上限の値 = 1.24/2 = 0.62
画素値 255 に対応する CT 値の下限の値 = 1.24*254.5 = 315.58
を除外した範囲だけを描けばヒストグラムの見栄えが良くなる：

```
plot_a.csh ./byte_a.txt 0.62 315.58 > byte_a.ps
```

このようにして作成した画像と図の出来栄えを確認する：

```
% xv byte_a.tif
% evince byte_a.ps
```

[12] 位相 CT 画像に対する後処理

位相 CT 画像に対する後処理の手順は吸収 CT 用のものと概ね同じである：

```
% mm.csh tg_p
    -23.171337      24.950184

16 ビット画素値１階調ごとの CT 値の増分の値
    = (24.950184-(-23.171337))/(2^16-1) = 0.0007342873426413

% mm.csh tg_p 16
    -23.171337      24.950184       0.0007342873426413

% mkdir word_p
% t2t_f.csh tg_p   -23.171337 0.0007342873426413 16 word_p > word_p.txt

% t2t_float tg_p - -23.171337 0.0007342873426413 16 -      > word_p.txt

% plot_x11.csh word_p.txt

% plot_x11.csh word_p.txt 0 15      # CT 値 = 0 ～ 150 の部分を表示
```

測定 170516c の位相 CT 画像のヒストグラムのピークの CT 値 ≒ 7.52 (1e-6)
→ byte 画像の８ビット画素値１階調ごとの CT 値の増分 = 0.0752 (1e-6)

```
% mkdir byte_p
% t2t_f.csh tg_p 0 0.0752 8 byte_p > byte_p.txt
```

横方向優先の順番で５×３のマトリックス状に並べる場合

```
% pile_h.csh byte_p 117 24 15 5 3 byte_p.tif
```

縦方向優先の順番で３×５のマトリックス状に並べる場合

```
% pile_v.csh byte_p 117 24 15 3 5 byte_p.tif
```

```
% plot_p.csh ./byte_p.txt > byte_p.ps
```

byte 画像の画素値０に対応する CT 値の上限 = 0.0752/2 = 0.0376
byte 画像の画素値 255 に対応する CT 値の下限 = 0.0752*254.5 = 19.1384

```
% plot_p.csh ./byte_p.txt 0.0376 19.1384 > byte_p.ps

% xv byte_p.tif
% evince byte_p.ps
```

[13] 長さと表示輝度のスケールバーを付けた吸収と位相 CT 画像の作成（オマケ）

吸収と位相 CT 画像の両方に使える長さが 25 μm のスケールバーの画像の作成：

```
% bar_ls.csh 250 5 50 '' '25 um' ls.tif
```

吸収と位相 CT のそれぞれの byte 画像の値に応じた表示輝度のスケールバー：

```
% bar_gs.csh 750 25 50 0 "`echo 1.24¥*255 | bc -l` (1/cm)"  bar_a.tif
% bar_gs.csh 750 25 50 0 "`echo 0.0752¥*255 | bc -l` (1e-6)" bar_p.tif
```

スケールバーの画像と吸収 CT の byte 画像を並べた画像を張り合わせる：

```
% pile_gray 2 1 -250 0   255 bar_a.tif    bar_a.tif  ls.tif
% pile_gray 1 2 0    -50 255 browse_a.tif byte_a.tif bar_a.tif
```

スケールバーの画像と位相 CT の byte 画像を並べた画像を張り合わせる：

```
% pile_gray 2 1 -250 0   255 bar_p.tif    bar_p.tif  ls.tif
% pile_gray 1 2 0    -50 255 browse_p.tif byte_p.tif bar_p.tif
```

作成した吸収と位相 CT の browse 画像の確認：

```
% xv browse_a.tif browse_p.tif
```