```
sixm/sixm_recipe.txt : Sat Apr 23 16:02:53 JST 2016

SIXM の画像再構成などの手順

---------------------------------------------------------------
(0) 書庫ファイルのダウンロードと展開

    wget http://www-bl20.spring8.or.jp/\~sp8ct/tmp/sixm.taz
    →  ...

    tar xzf sixm.taz    # ディレクトリ sixm が展開される。
    rm sixm.taz         # 書庫ファイルは最新のものを使って下さい。

(1) プログラムとのインストールと C-shell scripts の実行の準備

    cd sixm
    make                # SIXM 用の C 言語プログラムのコンパイル
    →  ...

    csh install.4csh    # SIXM 用の C-shell scripts が使うものなどの準備
    →  ...

    pwd                 # ディレクトリ sixm のフルパス名を調べておく
    →  /パス名/sixm     # このフルパス名を SIXM_FULL_PATH と書く

    cd ..

(2) ディレクトリ sixm のフルパス名を log-in shell の設定ファイルに登録

#   log-in shell が csh もしくは tcsh の場合、
#   以下の１行を設定ファイル ~/.login に追加する。

        set path=(SIXM_FULL_PATH $path)

#   log-in shell が bash の場合、
#   以下の１行を ~/.bash_profile もしくは ~/.profile に追加する。

        PATH=SIXM_FULL_PATH:$PATH

#   なお、これらの実行パスの設定は log-in し直さないと有効にならない。

---------------------------------------------------------------
(3) 測定ごとのディレクトリを作成し、その中ですべての処理を行う。

    mkdir 151130n
    cd 151130n

#   SIXM の測定で得たファイルをここにコピー（リンク）する必要はない。
#   測定 151130n のファイルがディレクトリ /151130n/ の下にあるとする。

(4) SIXM の測定のログファイル（a.log）のチェック

    head -1 /151130n/a.log
    →  326,751,100,1,0          # scans = 326
                                # views = 751
                                # darks = 100

(5) SIXM のデータファイル（a.HIS）のチェック

    du -b /151130n/a.HIS        # バイト単位のファイルサイズを表示
    →  31366209536     /151130n/a.HIS

    expr 326 \* 751 + 100       # refraction profile（RP）の画像の総数を計算
    →  244926

    check_his /151130n/a.HIS | uniq -c
    →  31366206065                      # ファイル上の有効なデータのバイト数
       244926  128     500     2        # RP 画像の枚数 = 244926
                                        # RP 画像の横画素数 = 128
                                        # RP と投影画像の縦画素数 = 500
                                        # RP 画像のタイプ = 2

#   HIS 形式ファイルのサイズの食い違いは致命的ではない。
#   ファイル上の RP 画像の枚数が計算値と一致しない場合は、...

(6) SIXM のパラメータファイル（a.par）のチェック

    sed -n -e 9,12p -e 37p /151130n/a.par
    →  "87.5"          # ９ 行目：Dr = 87.5 nm = 87.5e-7 cm
       "87"            # 10 行目：Dz = 87 nm
       "6000"          # 11 行目：SDD = 6000 mm = 6000e-3 m
       "4.98"          # 12 行目：Dp = 4.98 um
       "16"            # 37 行目：     * 16 = 79.68 um

#   これらはログノートの値と違っていた（以下の処理ではログノートの値を使う）。

#       投影画像の画素の横幅、Dr = 100 nm = 100e-7 cm
#       投影画像の画素の縦幅、Dz = 109.3 nm
#       サンプルと検出器の距離、SDD = 6195 mm = 6195e-3 m
#       RP 画像の画素の横幅、Dp = 6.4935 um * 16 = 103.896 um

(7) 投影画像の作成

#   サンプル回転角が０度の投影画像 r[i,r]_0.tif だけを取り出す場合

        his2raw_B /151130n/a.HIS 326 0 100 ri_0.tif rr_0.tif
        →       0  390.323438  3988.388437  43.814417  83.537854

#       とりあえず見るだけなら RAW_B 形式の投影画像で十分だと思われる。
#       ログファイルから得た scans = 326 と darks = 100 をここで指定。

#   SIXM の画像の再構成処理には投影画像すべてを取り出しておく必要がある。

        mkdir ri rr
        his2raw_F /151130n/a.HIS 326 - 100 ri/%03d.tif rr/%03d.tif > raw.log

#       RAW_W 形式で精度的には問題ないが、念のため RAW_F 形式画像にした。
#       views = 751 なので個々のファイル名になる投影番号は３桁以内（%03d）。
#       この処理には数分を要するかもしれない。

#   ディレクトリ /151130n/ の下の測定ファイルはこれ以降の処理では使用しない。

---------------------------------------------------------------
(8) RI（raw intensity）の投影画像の観察

#   改造版の xv などを使って RI の投影画像のうちの何枚かを眺める。

        xv ri/*00.tif   # 100 枚ごとの RI の画像を表示

#   サンプル像の上下端のおおよその位置（y 座標値 y1 と y2）を調べる。
#       保持具などを含まないサンプル像の影が濃い部分だけを対象とする。
#       これらの座標値はサンプル回転中心の位置の推定処理で使う。
#       測定 151130n では y1 = 125 かつ y2 = 360 だった。
```

(9) RR（raw refraction）の投影画像の観察

#　サンプル像の外形を識別しやすい RR の投影画像を xv などで眺める。

```
xv rr/*.tif     # RR の投影画像すべてを表示
```

#　サンプル像の左右の画像端にある空気の部分の幅の最小値 L と R を調べる。
#　　L と R の値は５画素幅程度の精度で決め、10 以下の値は０と見なす。
#　　L と R の一方が０でもかまわないが、両方が０ではダメ（後述）。
#　　投影画像すべてで０でない L もしくは R の値であれば lucky です。
#　　測定 151130n では投影画像すべてで L = 25 かつ R = 55 だった。

(10) RR の sinogram の画像の作成と観察

#　投影画像から両方が０でない L と R の値を決めることができた場合も
#　念のため RR の sinogram の画像を作成してそれらを確認した方が良い。

#　投影画像すべてに対して両方が０でない L と R の値がない場合、
#　投影番号ごとに両方が０ではない L と R の値を決めないとダメだが、
#　その処理には投影画像を「縦切り」にした sinogram の画像を使うと便利。

```
mkdir rr_sg
echo - 0 0 | ri2sg_B rr - - rr_sg/%03d.tif > /dev/null
```

#　　RR の値をそのまま使うので L と R の値が０の VLR_file を指定する。
#　　SIXM の装置定数の指定が不要な RI 用のプログラム ri2sg_B を使う。
#　　観察するだけなので RAW_B 形式の sinogram の画像を作成すれば良い。
#　　投影画像の縦画素数が 500 なので sinogram の枚数は３桁以内（%03d）。

#　RR の sinogram の画像を観察して投影番号ごとの L と R の値を決める。

```
xv rr_sg/*.tif  # RR の sinogram すべてを表示
                # これらの画像の x 座標値 = 投影画像の x 座標値
                # y 座標値 = 投影番号（上端が０度、下端が 180 度）
```

#　視野からサンプル像がはみ出ていた測定の投影番号ごとの L と R の値の例
#　　測定 151130f
#　　　投影番号の範囲　　L　　R
#　　　-79　　0　　69
#　　　80-95　　0　　41
#　　　96-223　　34　　0
#　　　224-　　66　　0
#　　測定 151201b
#　　　投影番号の範囲　　L　　R
#　　　-339　　0　　15
#　　　340-544　　24　　15
#　　　545-　　24　　0

---

(11) SP（sum of projection）の画像の確認

#　X 線投影値の SP の画像の作成

```
echo - 25 55 | ri2sp ri - - sp_a.tif > sp_a.log
```

#　X 線位相値の SP の画像の作成

```
echo - 25 55 | rr2sp rr - - 1 1 sp_p.tif > sp_p.log
```

```
# SP の値の絶対値ではなく空間分布を見たいだけなので、
# ここでは SIXM の装置定数として SDD = Dp = 1 を指定した。
```

#　これらの画像を観察して SP が横（投影番号）方向に一定値かどうかを確認する。

```
xv sp_a.tif sp_p.tif
or      xv -preset 4 sp_a.tif sp_p.tif          # 擬似カラー表示

# サンプル像と空気の境界線が水平でないなら、
# 測定中にサンプルが鉛直方向にクリープした可能性がある。
```

(12) サンプルの回転中心の位置（センター値）の推定

#　サンプルの回転角が０度と 180 度の X 線投影値の投影画像を用いた推定

```
echo - 25 55 | ri2rc ri - - 125 360 0.25 0.25 ri2rc.tif
→       121.5   202.5   146     # センター値の推定値 r0 = 146
        -59     59      4       # 投影画像の縦ズレ量 dz = 4
```

#　サンプルの回転角が０度と 180 度の微分位相値の投影画像を用いた推定

```
echo - 25 55 | rr2rc rr - - 125 360 0.25 0.25 rr2rc.tif
→       121.5   202.5   146.5   # r0 = 146.5
        -59     59      2       # dz = 2
```

#　RMSD の画像を観察してその空間分布からセンター値の推定精度を評価する。

```
xv ri2rc.tif rr2rc.tif
or      xv -preset 4 ri2rc.tif rr2rc.tif        # 擬似カラー表示
```

#　通常はここで推定した２個のセンター値のうちで縦ズレ量が小さい方の値を
#　吸収と位相の両方の画像再構成処理で指定すれば良い（今の場合は 146.5）。

---

(13) センター値を変えた画像再構成のテスト

#　推定した２個のセンター値の差が１（画素幅）以上ある場合や
#　縦ズレ量の絶対値が２（画素幅）以上の場合はもちろんのこと、
#　そうでない場合も念のために画像再構成のテストを行う方が良い。

#　　ここでは測定 151130n のスライス番号 250 の吸収と位相の画像の
#　　それぞれを 146.5 − 6 〜 146.5 + 6 の範囲で 0.5 刻みで変えた
#　　合計 25 通りのセンター値のそれぞれを指定して再構成してみる。

#　吸収画像の再構成のテストの実行とその結果の観察

```
mkdir test_a
( echo - 25 55 ; echo \$ ) | \
test_a.csh ri - - 250 100e-7 140.5 152.5 test_a >>! test_a.log

xv test_a/250_*.tif             # センター値を変えた吸収画像の観察
or      xv -hist test_a/250_*.tif       # ヒストグラム平滑化で表示輝度を強調
```

#　位相画像の再構成のテストの実行とその結果の観察

```
mkdir test_p
( echo - 25 55 ; echo \$ ) | \
test_p.csh rr - - 250 6195e-3 103.896 140.5 152.5 test_p >>! test_p.log

xv test_p/250_*.tif             # センター値を変えた位相画像の観察
or      xv -hist test_p/250_*.tif       # 表示輝度を強調
```

#　テストの結果の再構成画像をマトリックス状に配置した画像を作成し、
#　eog（eye of GNOME）のような縮小表示が得意な画像ビュワーを
#　使って全貌を概観するのが良いかもしれない。

```
        pile_h.csh test_a 1 1 25 5 5 test_a.tif > /dev/null
        pile_v.csh test_p 1 1 25 5 5 test_p.tif > /dev/null

        eog test_a.tif test_p.tif

#   いずれにせよ、再構成画像に偽像が発生していないセンター値を探し出す。

(14) 本番の画像再構成

#   マルチスレッドで再構成するには事前にスレッド数（例えば４）の設定が必要。

        setenv THREADS 4        # log-in shell が csh もしくは tcsh の場合
        export THREADS=4        # log-in shell が bash の場合

#   吸収画像の再構成

        mkdir tg_a
        ( echo - 25 55 ; echo \$ ) | \
        ri2tg_W ri - - 100e-7 146.5 0 tg_a/%03d.tif > tg_a.log

#   位相画像の再構成

        mkdir tg_p
        ( echo - 25 55 ; echo \$ ) | \
        rr2tg_W rr - - 6195e-3 103.896 146.5 0 tg_p/%03d.tif > tg_p.log

-----------------------------------------------------------------
(15) word 画像とその画素値ヒストグラムの作成

#   吸収画像

        mkdir word_a
        tg2tg tg_a - word_a | tr \\t , > word_a.csv
        →       -608.053955     3031.351318     # CT 値（LAC）の最小値と最大値

#   位相画像

        mkdir word_p
        tg2tg tg_p - word_p | tr \\t , > word_p.csv
        →       -9.801408       36.524204       # CT 値（RID）の最小値と最大値

(16) word 画像のスライス画像をマトリックス状に配置した画像の作成

#   まず、スライス画像をマトリックス状に配置した画像を作成する。
#       測定 151130n の word 画像のスライス番号が 125 から 10 刻みの
#       合計 25 枚のスライス画像を５×５のマトリックス状に配置してみる。

        pile_h.csh word_a 125 10 25 5 5 word_a.tif > /dev/null
        pile_v.csh word_p 125 10 25 5 5 word_p.tif > /dev/null

#   その後、その画像の下部に２個のスケールバーの画像を貼り付ける。
#       長さのスケールバーの画像作成のために指定したパラメータ
#           バーの横幅、width（画素幅）= 10000 (nm) / 100 (nm) = 100
#           バーの縦幅、height（画素幅）= 0.025 * width
#           フォントのサイズ、size（画素幅）= 32
#           バーの左端に印字するラベル、left = ""（なし）
#           バーの右端に印字するラベル、right = "10 (um)"
#       グレースケールバーの画像作成のために指定したパラメータ
#           width = 300
#           height = 0.075 * width
#           size = 32
#           left = CT 値の最小値
#           right = "CT 値の最大値 (CT 値の単位)"
#       ２個のスケールバーの画像を 128 画素幅の間隔を空けて左右に並べ、
#       その画像の上部にスライス画像をマトリックス状に並べた画像を
#       64 画素幅の間隔を空けて配置した画像を作成する。

        bar_ls.csh 10000/100 -0.025 32 "" "10 (um)" 16 ls.tif

#       吸収と位相画像用の長さのスケールバーの画像（ls.tif）は同じもので OK。

        bar_gs.csh 300 -0.075 32 -608.053955 "3031.351318 (1/cm)" 16 bar.tif
        pile_gray 2 1 -128 0 65535 bar.tif ls.tif bar.tif
        pile_gray 1 2 0 -64 65535 word_a.tif word_a.tif bar.tif

        bar_gs.csh 300 -0.075 32 -9.801408 "36.524204 (1e-6)" 16 bar.tif
        pile_gray 2 1 -128 0 65535 bar.tif ls.tif bar.tif
        pile_gray 1 2 0 -64 65535 word_p.tif word_p.tif bar.tif

#       グレースケールバーの画像（bar.tif）は横並べの作業用にも使い回した。

        rm ls.tif bar.tif

#       スケールバーの画像は不要なので、最後にそれらのファイルを消去。

(17) word 画像の代表的な点を通る３断面と画素値ヒストグラムを並べた図の作成

#   word 画像を観察するなどして、その上の代表的な点の位置を決める必要がある。
#       測定 151130n ではその点の座標値を x = 129、y = 151、z = 250 とした。

#   図中に測定番号を埋め込むため、draw_[a,p].csh の PS 出力を置換する。
#       draw_a.csh word_a ... | sed s@word_a@151130n/word_a@g
#       draw_p.csh word_p ... | sed s@word_p@151130n/word_p@g

#   draw_[a,p].csh で描いた図の上部に２個のスケールバーを貼り付ける。
#       長さのスケールバーの作成のために指定したパラメータ
#           width（pt.）= 10000（nm）* (3*72)（pt.）/ {（109.3*500)（nm）}
#           height（pt.）= 0.025 * width
#           size（pt.）= 9
#           left = ""（なし）
#           right = "10 um"
#       グレースケールバーの作成のために指定したパラメータ
#           width = 1.25 * 72
#           height = 0.075 * width
#           size = 9
#           left = CT 値の最小値
#           right = "CT 値の最大値 (CT 値の単位)"

#   ２個のスケールバーを 27 pt. の間隔を置いて横に並べ、
#   その下に 18 pt. の間隔を置いて３断面とヒストグラムの図を配置する。

        ( ( bar_ls.csh "10000*3*72/(109.3*500)" -0.025 9 "" "10 (um)" ; \
            bar_gs.csh "1.25*72" -0.075 9 -608.053955 "3031.351318 (1/cm)" ) | \
          epspile -n 2 1 -o 27 0 ; \
          draw_a.csh word_a 129 151 250 100/109.3 | \
          sed s@word_a@151130n/word_a@g ) | gspile -n 1 2 -o 0 18 > word_a.ps

        ( ( bar_ls.csh "10000*3*72/(109.3*500)" -0.025 9 "" "10 (um)" ; \
            bar_gs.csh "1.25*72" -0.075 9 -9.801408 "36.524204 (1e-6)" ) | \
          epspile -n 2 1 -o 27 0 ; \
          draw_p.csh word_p 129 151 250 100/109.3 | \
          sed s@word_p@151130n/word_p@g ) | gspile -n 1 2 -o 0 18 > word_p.ps
```

```
------------------------------------------------------------
(18) byte 画像作成のための８ビット画素値１階調ごとの CT 値の増分の決定
(19) byte 画像とその画素値ヒストグラムの作成
(20) byte 画像のスライス画像をマトリックス状に配置した画像の作成
(21) byte 画像の代表的な点を通る３断面と画素値ヒストグラムを並べた図の作成
```