

**高分解能 X 線 CT 装置**

**画像再構成手順書**

**JASRI / SPring-8**

**2004/7/4 版**

再構成手順(画像再構成パッケージの取り扱い方法) .....	3
パッケージの構成 .....	3
画像再構成(実践編) .....	4
各プログラムの説明 .....	15
画像再構成の原理 .....	30

## 再構成手順(画像再構成パッケージの取り扱い方法)

CT の撮影が完了したら、画像再構成を行う。その際に必要なのは、試料透過像・入射光・暗電流データ・撮影ログ、である。撮影マニュアルに従ってファイル名をつけたとすると、それぞれ q001.img q002.img ..., d01.img d02.img, output.log などとなっているはずである。q???img は投影数などにより変化するので、その都度異なったファイル数になる。もしデータセットに不安があるときは output.log と整合が取れているか調べればよい(整合が取れていないときは、再度撮影するのが最も安心)。また、データ取得を ImagePro で行ったり、HiPic でも間違えて tiff 形式で保存してしまった場合、ファイルは q001.tif q002.tif ..., d01.tif d02.tif, output.log などとなっている。この場合でも画像再構成は本パッケージで可能である。

### パッケージの構成

本パッケージには、“ct\_cbp.exe” “ct\_sino.exe” “ct\_sino\_tif.exe” “rec2tif.exe” “sino2tif.exe” “img\_ave.exe” “img2tif.exe” “sfa.exe” “srec.exe” “tif\_h2o.exe” “tif2hst.exe” という実行形式(exe ファイル)と “mkdark.bat” “reconst.bat” “sinogram.bat” の3つのバッチファイルが含まれている(ほかにもいくつか入っているが、あまり気にしない)。すべての実行形式ファイル・バッチファイルをパスの通っているフォルダに入れておくことと便利である。各プログラムの説明は後述する。また、「**画像再構成(実践編)**」に従って、ダミーデータの画像再構成を行うと、雰囲気がかみやすいだろう。ダミーとは言っても、実際の撮影データである。

(注意) filtered back projection method のソフトである ct\_fbp.exe の配布は現時点(2004.2.4)では中止している。(計算結果がおかしいので)プログラム中にバグがあることが分かっているが、どこにどんな類のものが潜んでいるのか探るのが大変なため。

画像再構成は DOS 窓にて次のように進められる。ただし、暗電流データは各データセットにつき、一つだけ作ればよい。

- 0 . 暗電流データの決定(普通は2つのデータの平均をとるだけ)。
- 1 . ある層を再構成するために必要なデータ(シノグラム)を作る。
- 2 . 再構成演算(Convolution Back Projection method)
- 3 . 見やすい絵に変換(線吸収係数の分布を gray scale の一般的な画像にする)

それぞれの処理を個別に行うのは煩雑なので、通常はバッチファイルを作りそれを実行するようにしている。複数レイヤーの連続再構成も行えるようなソフトもパッケージされている。

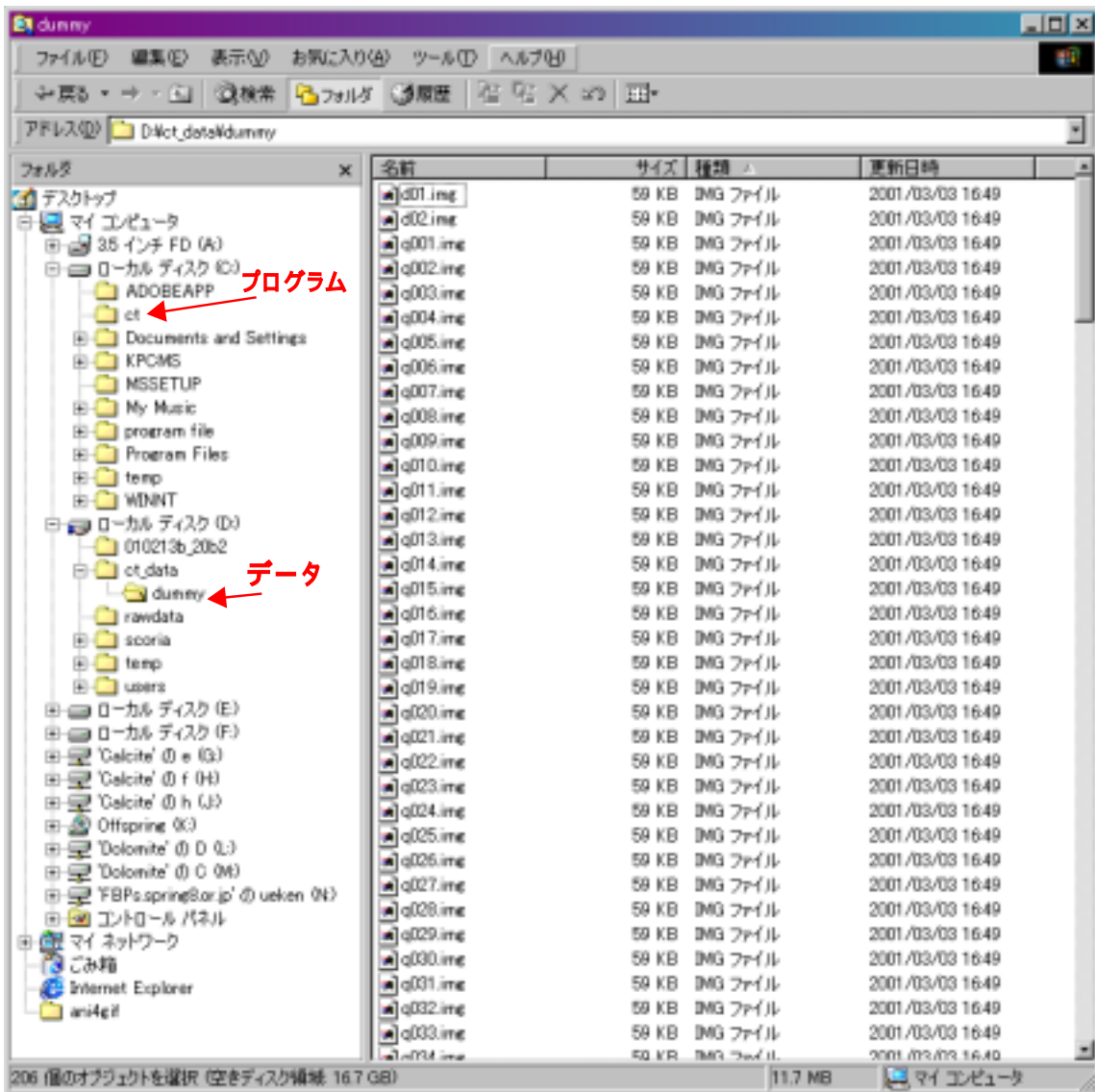
## 画像再構成(実践編)

ここではダミーデータである ct\_test.lzh を画像再構成することにより、本パッケージの使い方を習得する。そえるものは、windows マシン一台。ホームページからダウンロードしたプログラムおよび説明書き一式(ct\_manual.lzh)。これらは

<http://www-bl20.spring8.or.jp/xct/index.html>

に最新版がアップロードされているはずである。ダミーデータの情報は解凍時に生成される、ct\_test.txt というファイルに記述されている。

### a. 画像再構成プログラムおよびダミーデータの展開



ダウンロードしたプログラムおよび、ダミーデータを展開する。上の図ではプログラム・バッチファイルをすべて“c:¥ct”に、ダミーデータを“d:¥ct\_data¥dummy”に展開したところを示しているが、これはユーザー毎に任意の場所に展開してよい。ただし、プログラム・バッチファイルを展開したディレクトリにはパス(path)が通っていたほうが便利である。pathの通し方は、最後のほうに書かれている「pathの通し方」を参照。

## b. 暗電流データの作成

ダミーデータを展開すると、d01.img d02.img q001.img...q201.img output.log が一つのディレクトリに生成されているはずである(先の図に一部示されている)。これが出来ないときは、ダウンロードか lzh ファイルの解凍に失敗しているの、もう一度ダウンロードからやり直し。

次に、コマンドプロンプトを開き、図のようにカレントディレクトリの変更を行う。

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\>d:
D:\>cd ct_data\dummy
D:\ct_data\dummy>
```

```
D:\ct_data\dummy>mkdark
D:\ct_data\dummy>img_ave d01.img d02.img dark.img
read d01.img
read d02.img
output dark.img
D:\ct_data\dummy>
```

“mkdark.bat”というバッチファイルを用いて、画像再構成やシノグラム作成に用いる暗電流データを作る。下の図のように正常終了し、ディレクトリ内に”dark.img”が出来ていれば OK。

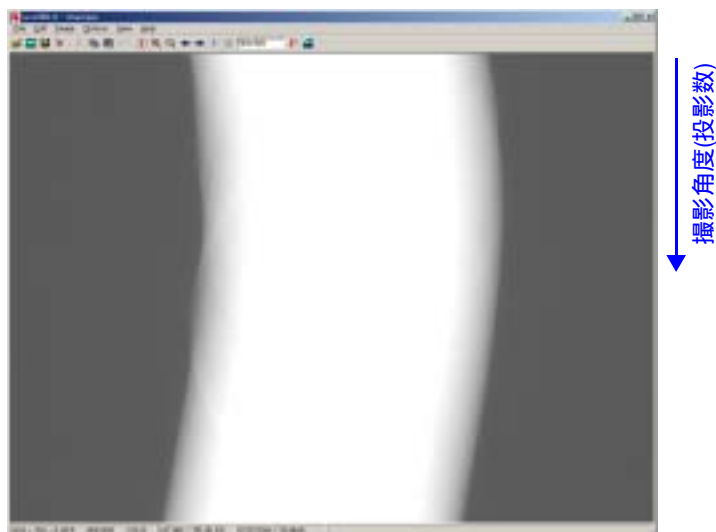
### c. シノグラムの作成

画像再構成する前にシノグラムの作成を行う。ただし、画像再構成に必須の手順ではないので省略しても良い。ダミーデータの画像サイズは 300(W) × 100(H) なので選べる層は 0 から 99 である。ここでは真中あたりの層(50 layer)の計算をさせることにする。

sinogram.bat を実行すると下の図のようになる。

```
D:\Yct_data\Ydummy\data>sinogram 50 ← sinogram.bat を実行
D:\Yct_data\Ydummy\data>ct_sino 50 q dark.img
rehot = 750, NIO = 151, total = 903
Store Sinogram 4.380000 / sec
D:\Yct_data\Ydummy\data>sino2tif
Image size: 1024 x 750
MAX and min: 1.639020 to -0.043540
CT image to: 16 bits TIFF.
Stored to: sino0050.tif
rotation axis: 501.455054
D:\Yct_data\Ydummy\data>
```

無事に "sino0050.tif" が生成されていることを確認したら、画像を表示してみる。下の図は IrfanView というフリーソフトを用いて表示した様子。この図では横軸は検出器の位置、縦軸は角度をあらわしており、一番上の層は"1°での投影像"を、一番下の層は"180°での投影像"をあらわしている。従って、縦軸の長さは投影数で決まる。このデータの場合は 0.24°ごとに 750 投影のデータを持っているので、縦方向の長さは 750pixel である。白い帯状のうねりが見えているが、これは"回転軸"と"円柱の中心"の位置がずれているためでバグではない。このことは再構成像からも解るだろう。

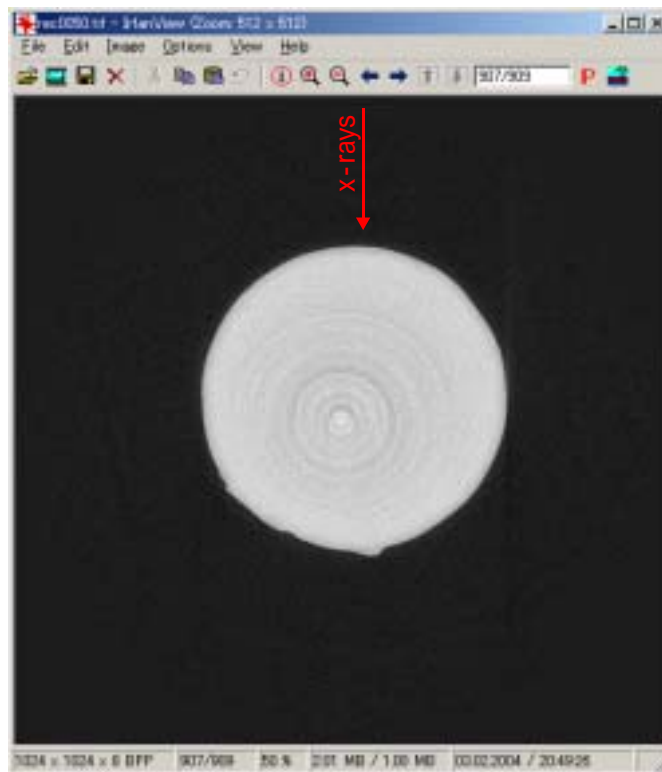


d. 画像再構成

いよいよ画像再構成を行う。ここでもシノグラムの時と同様、真中あたりの層(50 layer)の計算をさせることにする。

```
コマンド プロンプト
D:\Yct_data\Ydummydata>reconst 50 ←reconst.bat を実行
D:\Yct_data\Ydummydata>ct_sino 50 a dark.img
rehot = 750, N10 = 151, total = 903
Store Sinogram 0.796000 / sec
D:\Yct_data\Ydummydata>sino_conv
D:\Yct_data\Ydummydata>ct_cbp
calculation layer : 50
sinogram file name : s0050.sin
Width of image : 1024
Number of projection : 750
Rotation center : 501.455054
Convolution : 2.391000 / sec
BackProjection : 14.438000 / sec
D:\Yct_data\Ydummydata>rec2tif 16
Image size: 1024 x 1024
MAX and min: 6.739176 and -0.839662
CT image to: 16 bits TIFF.
Conv. const.: 8647.235857 -0.839662
pixel size: 5.830000
Stored to: rec0050.tif
D:\Yct_data\Ydummydata>
```

無事に”rec0050.tif”が作成できたでしょうか？出来ていれば、IrfanView を用いて下のような図が表示できるはずである。



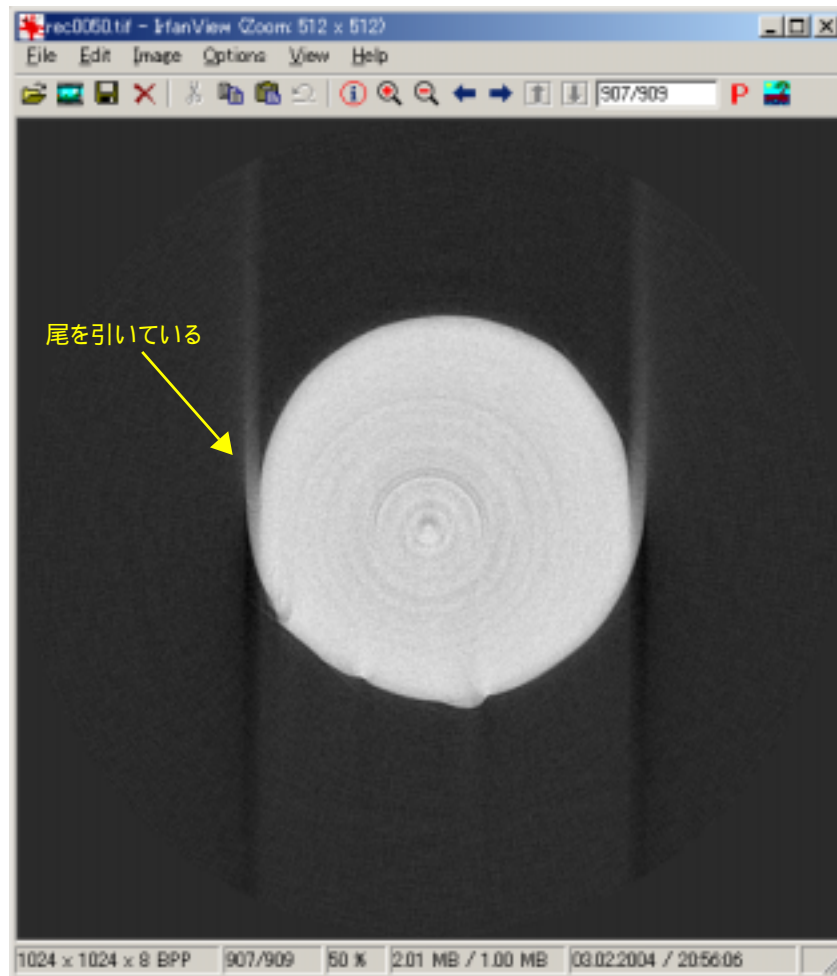
ここで簡単に再構成像について説明する。この図は試料を上から見た図であり、初期位置(角度 0° の場所)では X 線は図の上より入射される(図の矢印参照。検出器は下にいる事になる)。また、図の濃淡(画素値という)は X 線の線吸収係数(Linear Absorption Coefficient, LAC)に変換されうるもので、この場合は

$$\text{LAC} = \text{画素値} \div 8647.235857 + (-0.839662)$$

の式が成り立つ。8647.235857 と-0.839662 が conversion constants として計算時に表示されるし、TIFF のタグの中にも埋め込まれている。このような中途半端な値の変換係数になるのは、1 枚の CT 像中の最大と最小の吸収係数を機械的に 16bit の範囲に収めていることによる。

中央の試料の中にリング状に見えるムラは「ring artifact」と呼ばれる物で、検出器の感度ムラや入射ビームの揺らぎなどにより発生する。これを画像処理で消すのは大変困難で、透過像の撮影のときにきれいなデータを撮るようにするべきである。この試料の撮影をしたときは、工夫が十分ではなかった。

ところで、実際のデータを再構成すると、しばしば下の図のように試料のふちで尾を引いたような像を作ることがある。これは実際の回転軸の位置と、画像再構成の演算の際に用いた値がずれているために起こる偽像で、透過像に屈折コントラストが現れていたり、試料の透過率適切でない場合によく起こる。



実際の回転軸の位置と、演算で用いた回転軸の位置がずれているとき



この偽像はダミーデータを正しく演算していないときに生じたもので、回転軸を実際とは異なる値にして無理やり計算したものである。正しくはダミーデータの回転軸は 501.5 くらいであるが、この演算を行ったときには回転軸を 510.0 にして行った。その時の DOS 窓は次のようになっている。

```

コマンド プロンプト
D:\Yct_data\dummy\data>reconst 50 510.
D:\Yct_data\dummy\data>ct_sino 50 q dark.img
rshot = 750, N10 = 151, total = 903
Store Sinogram 0.719000 / sec

D:\Yct_data\dummy\data>sino_conv 510.
D:\Yct_data\dummy\data>ct_cbp
calculation layer      : 50
sinogram file name    : s0050.sin
Width of image        : 1024
Number of projection  : 750
Rotation center       : 510.000000
Convolution           : 2.407000 / sec
BackProjection        : 15.281000 / sec

D:\Yct_data\dummy\data>rec2tif 16
Image size:          1024 x 1024
MAX and min:         6.707672 and -1.349084
CT image to:         16 bits TIFF.
Conv. const.:        8134.290831  -1.349084
pixel size:          5.830000
Stored to:           rec0050.tif

D:\Yct_data\dummy\data>

```

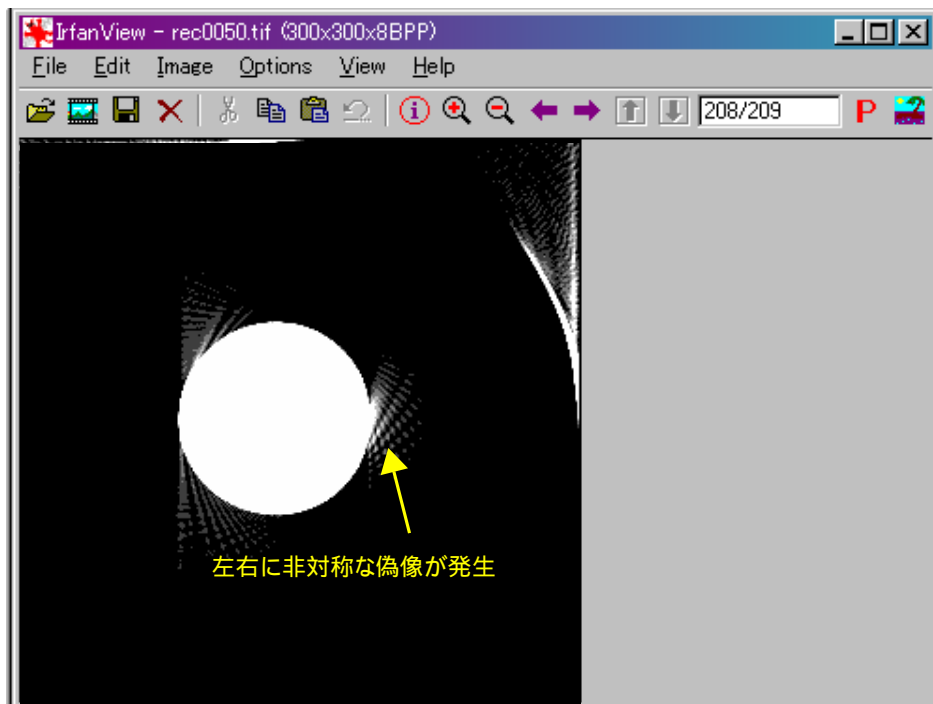
reconst.bat を実行  
回転軸を510.0に指定

回転軸を510.0と設定

計算誤差も大きい

回転軸の位置が正しくないと、偽像が発生し計算誤差も大きくなる。これでは LAC の定量的取り扱いが不可能になるので、注意が必要である。

また、測定時に何らかの原因により、試料が回転軸とは無関係に移動したときは、次の図のように像を結ばない。このときはいくら回転軸の位置を修正しても偽像が消えない。再測定が必要である。



左右に非対称な偽像が発生

測定中に試料が回転軸と無関係に動いたとき

この図はダミーデータを作成し、計算機の中で擬似的に試料を動かして作成した。

e. プログラムを実行したときに表示される値の意味

```
D:¥ct_data¥dummy>img_ave d01.img d02.img dark.img
```

```
read d01.img          d01.img を読み込む
read d02.img          d02.img を読み込む
output dark.img       dark.img を作成
```

```
D:¥ct_data¥dummy>ct_sino 50 q dark.img
```

```
nshot = 180, NIO = 19, total = 201      180 投影・I019 枚・合計 201 枚撮影
store sinogram 0.953000 / sec           シノグラムを保存
```

```
D:¥ct_data¥dummy>sino2tif 8
```

```
Image size:          300 x 180          像のサイズ 300(W) × 180(H)
MAX and min:         0.580355 to 0.000000 像の中の最大値・最小値
CT image to:         8 bits TIFF.        TIFF の bit 数
Stored to:           sino0050.tif        出力ファイル名
Rotation axis:      501.455054          重心演算で求めた回転軸の位置
```

```
D:¥ct_data¥dummy>sino_conv 510.
```

回転軸の位置を 510. に強制変更

```
D:¥ct_data¥dummy>ct_cbp
```

```
Width of image      :          300      像の横幅
Number of Projection :          180      投影数
Rotation center     : 151.0000000      回転軸の位置
reconstruction      :          4 / sec  再構成に要した時間
```

```
D:¥ct_data¥dummy>rec2tif
```

```
Image size:          300 x 300          再構成像の出力サイズ
MAX and min:         10.113399 and -3.240130 像の中での LAC の最大・最小値
CT image to:         16 bits TIFF.        TIFF イメージの bit 数
Conv. const.:       10      0.000000      LAC と画素値の変換係数
Pixel size:         5.830000            ピクセルサイズを 5.83µm とした
Stored to:          rec0050.tif          出力ファイル名
```

f. 回転中心の決定

実際の試料を再構成する場合、重心演算で自動的に求めた回転軸の位置が実際とは異なることが多い。重心演算を信頼しすぎると、間違ったまま解析を進めてしまいかねない。結局最後には自分の目で見て判断するしかない。そのために少しずつ回転軸の値を変化させながら再構成を行う必要がある。

練習のためダミーデータで実行する。

```
D:¥ct_data¥dummy>sfa.exe 50 499. 503. 1.0 5.83 > 50.bat
```

このように実行すると、"50.bat"というバッチファイルが作成される。

コマンドの意味は、「50 レイヤーを、回転軸の位置を 499 から 503 まで 1.0pixel ずつ移動させて画像を再構成させるバッチファイルを作る。ファイル名は"50.bat"で、1pixel のサイズは 5.83 $\mu$ m である。」詳細はコマンドのリファレンスに記述されている。

作成されたバッチファイルの中身は次のとおり

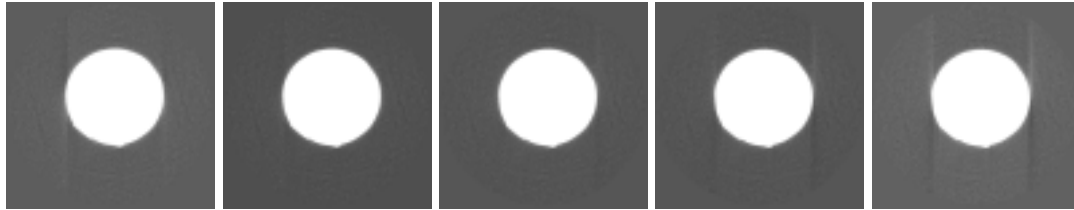
\*\*\*ここから\*\*\*

```
ct_sino 50 q dark.img          50 レイヤーのシノグラムを作成
copy s0050.sin tmp.sin        s0050.sin を tmp.sin にコピー
sino_conv 499.000000          回転軸の位置を 499.0 に設定
ct_cbp                         CBP 法で画像再構成
rec2tif 16 5.830000           TIFF 形式に変換
ren rec0050.tif 0050_04990.tif rec0050.tif を 0050_04950.tif に名前を変更
copy tmp.sin s0050.sin        tmp.sin を s0050.sin にコピー
sino_conv 500.000000          回転軸の位置を 500.0 に設定
ct_cbp                         (以下省略)
rec2tif 16 5.830000
ren rec0050.tif 0050_05000.tif
copy tmp.sin s0050.sin
sino_conv 501.000000
ct_cbp
rec2tif 16 5.830000
ren rec0050.tif 0050_05010.tif
copy tmp.sin s0050.sin
sino_conv 502.000000
ct_cbp
rec2tif 16 5.830000
ren rec0050.tif 0050_05020.tif
copy tmp.sin s0050.sin
sino_conv 503.000000
ct_cbp
rec2tif 16 5.830000
ren rec0050.tif 0050_05030.tif
del tmp.sin                    tmp.sin を消去
***ここまで***
```

作成されたバッチファイルを実行する(速いマシンならば、数分で終わるはず)。

D:¥ct\_data¥dummy>50.bat

すると、0050\_04990.tif から 0050\_05030.tif まで、レイヤーと回転軸の位置をファイル名にもつー連の TIFF ファイルが出来上がる。下にそれぞれの再構成像をコントラストをエンハンスさせて示す。



この図では判りにくいですが、0050\_05010.tif が上下に尾を引いていない像になっている(自分で実行して確認すること)。0050\_04990.tif あるいは 0050\_05030.tif は試料の両サイドに上下に尾がひいているのが分かる。つまり、0050\_05010.tif が正しい像を示している。従って、50 レイヤーでの回転軸の位置は 501.0 近辺であることが判る。より詳細に調べるためには、ステップ幅を 0.5pixel 位にするとよい。しかし、あまり細かく刻んでも、意味無く時間ばかり消費することになる。

#### g. 連続再構成

3次元像を出力するためには、必要なレイヤーの画像再構成を行いスタックしなければならない。ここでは、`ct_sino.exe sino_conv.exe ct_cbp.exe rec2tif.exe` を用いて連続再構成をさせるバッチファイルを自動生成するプログラムの使用例を示す。

練習のためダミーデータで実行する。

```
D:¥ct_data¥dummy>src 5 501. 10 502. 5.83 > 5-10.bat
```

このように実行すると、“5-10.bat”というバッチファイルが作成される。この場合は、レイヤー5での回転軸の位置は 501 でレイヤー10での回転軸の位置は 502 である(`sfa.exe` を用いていくつかのレイヤーにおける回転軸の位置を決めておくことと便利)。ピクセルサイズは 5.83 $\mu\text{m}$ 。次に、生成されたバッチファイルを実行する。

```
D:¥ct_data¥dummy> 5-10.bat
```

そうすると、`rec0005.tif` から `rec0010.tif` までの再構成像が 16bit tiff 形式で生成される。これも速いマシンならば、数分で終わるはず。

#### h. 再構成画像の 8bit 化

ここまでの再構成画像はすべて 16bit tiff 形式での出力だが、実際の画像処理や 3次元像を作成するときには、8bit 画像にしておいたほうが無難である。さらに、市販のソフトなどで画像処理あるいは画像表示をするために、全ての再構成画像のコントラストを揃えておいたほうがよい。次に、`rec2tif` で出力した 16bit の再構成画像(`rec????.tif`)を、8bit tiff 画像(`ro????.tif`)に変換するソフトの使用法を示す。

```
D:¥ct_data¥dummy>tif_h2o 100 200 0.0 2.0
```

このコマンドを実行すると、`rec0100.tif ~ rec0200.tif` が `ro0001.tif ~ ro0101.tif` に変換される(始めの二つの引数で、変換開始レイヤーと変換終了レイヤーを指定する)。このとき、`rec0100.tif` が `ro0001.tif` に変換されるので、注意が必要である。次の二つの引数は、8bit でどこからどこまでの吸収係数を表示するかを決める。例の引数で実行すると `ro????.tif` の画素値 0 が 0.0 $\text{cm}^{-1}$  で画素値 255

が  $2.0\text{cm}^{-1}$  となる。また、このソフトを用いると、変換と同時に特定領域の切り出しも行うことができる。

```
D:¥ct_data¥dummy>tif_h2o 100 200 0.0 2.0 100 150 250 350
```

と実行すると、rec0100.tif ~ rec0200.tif から(100,150)-(250,350)の領域が切り出され、ro0001.tif ~ ro0101.tif に  $0.0\text{-}2.0\text{cm}^{-1}$  で 8bit に規格化されたファイルとして出力される。

#### i. 再構成画像のヒストグラムを作る

8bit に変換するときの係数はどのように決めるのだろうか？これには明確な決め方は存在しない。それは、8bit に変換する目的に「見やすい絵が出力したい」とか「定量解析がしたい」などの複数の目的があるためである。しかし、闇雲にトライ&エラーで探すよりはある程度指標が合ったほうが便利である。最も簡単に思いつくのは、注目したい再構成像がどんな(線吸収係数の)ヒストグラムなのかを明らかにすることである。ここでは、16bit tiff 形式の再構成画像からそのヒストグラムを作成する方法を示す。

rec0050.tif という 16bit の再構成画像(7 ページ目で作成した画像)のヒストグラムを得るには、次のように入力する。

```
D:¥ct_data¥dummy>tif2hst rec0050.tif rec0050.hst
```

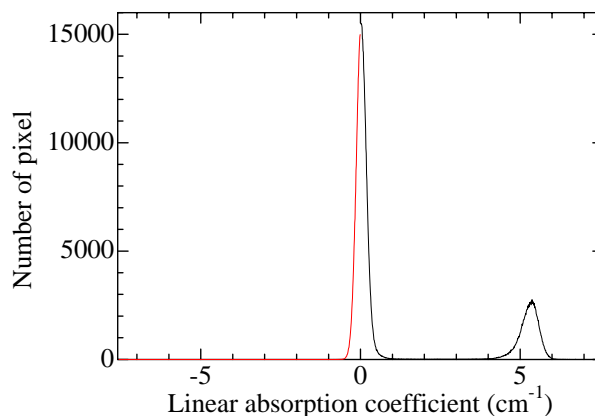
ただし、rec0050.hst というのは任意のファイル名でよい。rec0050.hst の中身はアスキー形式で、次のようになっている。任意のグラフソフトで描画することが可能で、右下の図のようになる。

\*\*\*rec0050.hst の中身\*\*\*

0.010000	15510	14990
0.020000	15468	14883
0.030000	15482	14551
0.040000	15428	14240
0.050000	15310	13806
0.060000	14795	13163
0.070000	14769	12964
0.080000	14240	12121
0.090000	14180	11780
0.100000	13419	10971

(以下省略)

\*\*\*\*\*



rec0050.hst ファイルは左のカラムから順に、吸収係数( $\text{cm}^{-1}$ )・一番目のカラムが正の場合の画素数・一番目のカラムが負の場合の画素数、となっている。右上のグラフで言うと、X軸が一番目のカラムで、黒い曲線が二番目のカラム。赤い曲線が三番目のカラムである。

0cm<sup>-1</sup>周辺のピークは空気のもので、5cm<sup>-1</sup>近辺のピークは試料によるものである。これは実際にシリカガラスを撮影した物で、有限の投影数と有限の空間分解能の検出器を使っているために、計算誤差が発生していることも表している。ちなみに照射 X 線も少しは揺らいでいる。

複数のレイヤーに関してのヒストグラムを得たいときには、複数の再構成画像を作成してからバッチ処理などを行い表計算ソフトやグラフソフトで積算する。

#### j. TIFF 画像の表示

この説明書では TIFF の画面出力に IrfanView というフリーソフトを用いた。このソフトは多種多様な画像フォーマットに対応している。また、フォルダ内の画像データの連続再生などの機能も備えており、使い勝手がよい。CT 像を連続して見たい場合などに非常に便利である。このソフトは <http://www.irfanview.com/> からダウンロード可能。

また、フリーソフトで 16bit tiff 形式の画像処理が可能な Image Tool というソフトもある。マクロあるいはプラグインによる機能の拡張もできる、かなり高機能なソフトである。詳細は <http://ddsdx.uthscsa.edu/dig/itdesc.html> を参照のこと。

#### k. その他

不備・不明な点があるときは

財団法人 高輝度光科学研究センター 利用研究促進部門 II  
上杉 健太郎  
[ueken@spring8.or.jp](mailto:ueken@spring8.or.jp)

まで、お問い合わせください。

## 各プログラムの説明

### 1. Windows 用実行形式(exe ファイル)

括弧なしの引数は必須パラメータである。括弧がついている引数はデフォルト値が設定されており、一応省略可能。しかし、種類によっては適切な値を入力しないと、**正しい結果を出さない**ので注意が必要である。たとえば、画像再構成の時には試料の回転軸位置の値は省略可能である。しかし、試料による X 線の吸収が少なかったり、屈折コントラストが付いている時には重心演算に失敗するので、正しい値を得ることはまれである。この場合は、回転軸の位置をシフトさせながら画像再構成を行い、再構成像(CT 像)を見ながら値を決めていくしかない。

#### a) `img_ave.exe`

機能 :

複数の `img` ファイル(HiPic により生成されるファイル)の平均を求め、ファイルに出力する。

記述 :

```
%img_ave dark1 dark2 dark
```

引数 :

**dark1** : 1 つめの暗電流データ(入力)

**dark2** : 2 つめの暗電流データ(入力)

**dark** : 上の二つの平均(出力)

使用例 :

```
1. d:¥data¥010226e> img_ave d01.img d02.img dark.img
```

```
2. d:¥data¥010226e> img_ave d01.img d02.img d03.img dark.img
```

意味 :

1. `d01.img` と `d02.img` のピクセル毎の平均値を `dark.img` に出力する。

2. `d01.img` と `d02.img` と `d03.img` のピクセル毎の平均値を `dark.img` に出力する。

## b) ct\_sino.exe and ct\_sino\_tif.exe

機能 :

実験により得られた、一連の透過像と入射光のデータから、CT 像再構成のために必要なデータの抽出し、シノグラムを作る。横軸は検出器の位置、縦軸は投影像の角度の情報(投影数)を持っている。

ct\_sino.exe は.img 形式の画像からシノグラムを作り、ct\_sino\_tif.exe は TIFF 形式の画像からシノグラムを作る。

記述 :

```
%ct_sino layer (head) (dark)
```

引数 :

layer : シノグラムを作りたい場所(layer)の値

head : 一連の透過像が格納されているファイルの先頭文字(デフォルトは"q")

dark : 暗電流のデータが入っているファイル名(デフォルトは"dark.img")

使用例 :

1. d:¥data¥010226e> ct\_sino 100
2. d:¥data¥010226e> ct\_sino 100 q d01.img
3. d:¥data¥010226e> ct\_sino 100 r d00.img

意味 :

1. 透過像で 100 番目の層のシノグラムを作る。第二・第三の引数は省略。これは、生データが"q001.img q002.img ..." という構成になっていて、暗電流のデータファイルが "dark.img" である時に可能。
2. 1.と同様に 100 番目のシノグラムを作る。1 との違いは暗電流データに "d01.img" を用いるように指定したこと。
3. この場合、暗電流データに "d00.img" を指定し、生データのセットが "r001.img r002.img..." というような構成になっている事をあらわしている。

備考 :

ct\_sino.exe はバイナリ形式の 2 次元データを出力する(s????.sin という layer 番号を反映した名前)。いくつかのパラメータを次に実行されるファイルに渡すために sino.tmp というファイルを作る。これは再構成演算後も残っているが、演算後に不要の場合は消去してもかまわない。ちなみに sino.tmp に入っている情報は画像の横サイズ・投影数・シノグラムの layer 番号・回転軸の位置(pixel 単位)の 4 つである。TIFF 画像をみればわかるが、サイン波の形をしていることから「シノグラム(sinogram)」と呼ばれている。

### sino.tmp の出力例

1000	横方向の画素数
720	投影数
0100	layer 番号
461.076695	重心演算による回転軸の位置



### c) ct\_cbp.exe or ct\_fbp.exe

機能 :  
ct\_sino.exe により抽出された layer のデータを用いて画像再構成を行う。

記述 :  
%ct\_cbp

引数 :  
なし

使用例 :

1. d:¥data¥010226e> ct\_cbp

意味 :  
1. ct\_sino.exe から引き渡された情報だけから、画像再構成を行う。この場合、再構成像のサイズは(シノグラムの横のサイズ)<sup>2</sup>となる。

備考 :  
ct\_fbp.exe はバイナリ形式(double 型)の 2 次元データを出力する(r????.rec という layer 番号を反映した名前)。いくつかのパラメータを次に実行されるファイルに渡すために rec.tmp というファイルを作る。  
このプログラムの詳細は RECLBL のマニュアルを参照のこと(<http://cfi.lbl.gov/>)。

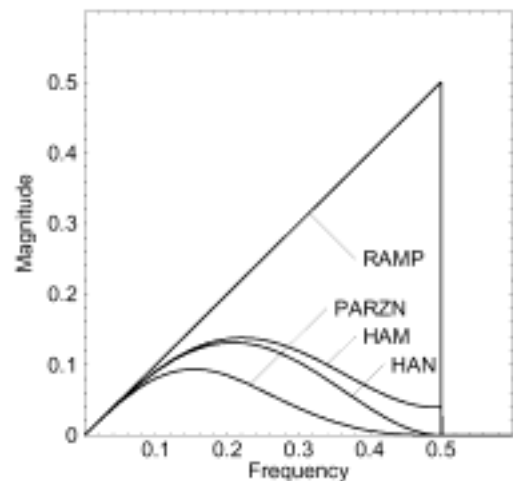
#### rec.tmp の出力例

0100	layer 番号
1000	シノグラムの横方向の画素数
720	投影数
900	再構成像の出力サイズ
461.0000	回転軸の位置
2	filter 関数の番号

#### filter 関数の種類

フィルター関数を変更したいときは、プログラムをコンパイルしなおす必要がある。

1. HAN
2. HAM
3. RAMP
4. PARZN
5. BUTER



ct\_cbp.exe ではフィルター関数は HAN 型に固定してある。これ以外のフィルター関数を用いる時は、再コンパイルが必要。

#### d) rec2tif.exe

機能 :

ct\_cbp.exe により作成され、double 型で出力された再構成像を 8bit or 16bit TIFF 画像に変換する。画像のサイズは ct\_cbp.exe が出力した再構成像の大きさと等しい(NDIMU × NDIMU)。

記述 :

%rec2tif (bit) (delta) (div) (base)

引数 :

**bit** : 8 or 16 bit を選択(デフォルトは 16bit)  
**delta** : pixel の間隔(ミクロン単位、デフォルトは 5.83μm)  
**div** : LAC 値と画素値の変換係数(傾斜)(bit と LAC による可変値、後述)  
**base** : 画素値 0 に対応する LAC 値(デフォルトは CT 像の中の最小値)

使用例 :

1. d:¥data¥010226e> rec2tif
2. d:¥data¥010226e> rec2tif 8 12.
3. d:¥data¥010226e> rec2tif 16 5.8 100 10.

意味 :

1. rec.tmp と再構成像の double 型から TIFF 16bit 形式に変換
2. TIFF 8bit 形式に変換。検出器の 1pixel のサイズは 12μm
3. TIFF 16bit 形式に変換。変換係数も指定している。

備考 :

ct\_cbp.exe が作成した rec.tmp に記述してある情報を、TIFF 画像のタグに埋め込んでいる。tif\_inf.exe で抽出可能。記述してある情報は順番に、“bit 数”・“横サイズ”・“縦サイズ”・“投影数”・“検出器 1 pixel のサイズ”・“変換係数(div)”・“変換係数(base)”・“回転軸の位置”・“フィルタ関数(1-5)”。また、画素値と LAC(Linear Absorption Coefficient、線吸収係数)の実験値の関係は次の式のようにになっている。

$$\text{LAC} = \text{画素値} \div \text{div} + \text{base}$$

## e) sino2tif.exe

- 機能 :  
ct\_sino.exe により作成され、double 型で出力されたシノグラムを 8bit or 16bit TIFF 画像に変換する。画像のサイズは、撮影時の横幅 × 投影数。
- 記述 :  
%sino2tif (bit)
- 引数 :  
bit : 8 or 16 bit を選択(デフォルトは 16bit)
- 使用例 :  
d:¥data¥010226e> sino2tif
- 意味 :  
double 型で出力されているデータを、16bit の TIFF 画像に変換
- 備考 :  
ct\_sino.exe が作成した sino.tmp に記述してある情報を、TIFF 画像のタグに埋め込んでいる。PhotoShop では、“ファイル情報”で読める。また、WZ Editor 等のエディタでも無理やり読むことは可能。記述してある情報は順番に、“bit 数”・“横サイズ”・“縦サイズ”・“投影数”・“変換係数(div)”・“変換係数(base)”・“重心演算で推測した回転軸の位置”。また、画素値とシノグラムの値の関係は次の式のようになっている。

$$\text{sinogram}(p=\ln(I_0/I)) = \text{画素値} \div \text{div} + \text{base}$$

## f) img2tif.exe

機能 : HiPic により作成された画像データを 8bit or 16bit TIFF 画像に変換する。サイズの変換は行わない。

記述 :  
%img2tif input.img output.tif from to bit

引数 :  
input.img : 入力ファイルを指定(???.img)  
output.tif : 出力ファイル名を指定  
from : 8bit に変換するときの変換係数  
to : 8bit に変換するときの変換係数  
bit : 8 or 16 bit を選択

使用例 :  
1. d:¥data¥010226e> img2tif q001.img q001.tif 0 12000 8  
2. d:¥data¥010226e> img2tif q001.img q001.tif 0 0 16

意味 :  
1. q001.img を 8bit TIFF 画像に変換。TIFF での画素値 255 は img での 12000。  
2. img を 16bit TIFF に変換。img は 14bit なので、変換係数を入力しても無視する。

備考 :  
8bit に変換するときに from と to を 0 と 0 で指定すると、img での最小値と最大値を from と to に自動的に割り当てる。

### g) sino\_conv.exe

- 機能 :  
ct\_sino により生成された sino.tmp 中の回転軸の値を書き換えるプログラム。
- 記述 :  
%sino\_conv AXIS
- 引数 :  
AXIS : 回転軸の位置
- 使用例 :  
d:¥data¥010226e> sino\_conv 501.2
- 意味 :  
sino.tmp に記述されている回転軸の位置の値を 501.2 に書き換える。
- 備考 :  
シノグラムの作成(ct\_sino.exe)と再構成演算(ct\_cbp.exe)の間に実行する必要がある。

#### h) sfa.exe or sfa\_tif.exe

機能 : あるレイヤーにおいて、回転軸の位置を変えながら画像再構成させるためのバッチファイルを作成するプログラム。

記述 :  
`%sfa layer from to step (delta) > output`

引数 :  
**layer** : 計算したいレイヤー  
**from** : 回転軸の初期位置(単位はピクセル)  
**to** : 回転軸の到達位置(単位はピクセル)  
**step** : ステップ間隔(単位はピクセル)  
**delta** : 画素サイズ(単位はミクロン)  
**output** : 出力先のバッチファイル名(標準出力)

使用例 :  
`d:¥data¥010226e> sfa 10 148 152 1.0 5.83 > 20.bat`

意味 : **10** レイヤーを、回転軸の位置を **148** から **152** まで **1.0pixel** ずつ移動させて画像を再構成させるバッチファイルを作る。ファイル名は**"20.bat"**で、**1pixel**のサイズは**5.83 $\mu$ m**。

備考 : 1枚あたりの画像再構成の時間は短くならない。

### i) srec.exe or srec\_tif.exe

機能 : 複数の連続するレイヤーの再構成を行わせるバッチファイルを作成するプログラム。

記述 :  
%srec from axis1 to axis2 (delta) > output

引数 :  
from : 開始レイヤー  
axis1 : 開始レイヤーにおける回転軸の位置(単位はピクセル)  
to : 到達レイヤー  
axis2 : 到達レイヤーにおける回転軸の位置(単位はピクセル)  
delta : 画素サイズ(単位はミクロン)  
output : 出力先のバッチファイル名(標準出力)

使用例 :  
d:¥data¥010226e> srec 5 149 10 151 5.83 > 5-10.bat

意味 : 5 から 10 レイヤーまで連続再構成させるバッチファイルを作る。レイヤー5 での回転軸の位置は 149 で、レイヤー10 での回転軸の位置は 151 である。ファイル名は”5-10.bat”で、1pixel のサイズは 5.83 $\mu$ m。

備考 : 1 枚あたりの画像再構成の時間は短くならない。このプログラムを実行する前には、開始レイヤーと到達レイヤーでの回転軸の位置を決定しておく必要がある。sfa.exe 等を用いると比較的機械的に進められる。

## j) tif\_h2o.exe

機能 : 連続するレイヤーの再構成像(tiff 16bit)を tiff 8bit 形式に変換する。

記述 : %tif\_h2o sta\_ly dst\_ly LACmin LACmax (x1 y1 x2 y2)

引数 :

- sta\_ly : 開始レイヤー
- dst\_ly : 到達レイヤー
- LACmin : 8 bit 画像の画素値 0 の吸収係数(cm<sup>-1</sup>)
- LACmax : 8 bit 画像の画素値 255 の吸収係数(cm<sup>-1</sup>)
- x1 : 切り出し領域
- y1 :
- x2 :
- y2 :

使用例 :

1. d:¥data¥010226e> tif\_h2o 100 200 0.0 2.0
2. d:¥data¥010226e> tif\_h2o 100 200 0.0 2.0 100 150 200 350

意味 :

1. **100(rec0100.tif)**から **200(rec0200.tif)**レイヤーまでを **8 bit tiff** 形式の画像 **ro0001.tif** から **ro0101.tif** に変換する。このとき **ro????.tif** の画素値 **0** は吸収係数 **0.0cm<sup>-1</sup>** に相当し、画素値 **255** は吸収係数 **2.0cm<sup>-1</sup>** に相当するようにコントラストをそろえる。
2. 1 の動作とともに、16bit 画像からの切り出しを行う。(100,150)-(200,350)が切り出され、101x201 の画像サイズになる。

備考 :

このソフトでは、rec????.tif に埋め込まれている情報(吸収係数への変換係数)を基に画像を取り扱う。したがって、rec2tif.exe あるいは tif\_h2o.exe 自身によって作成された画像以外では正しい動作はしない。  
単に tiff 画像の 16bit から 8bit への変換がしたいときは、tif\_conv.exe を使うか何らかの画像変換ソフトを使わなければならない。



### k) tif2hst.exe

機能 : 再構成画像(tiff 16bit or 8bit)のヒストグラムを出力する。

記述 :  
%tif2hst input.tif outputfile

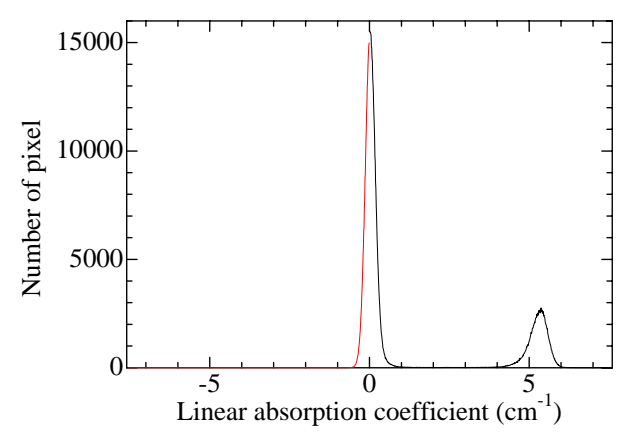
引数 :  
**input.tif** : ヒストグラムを作成したい再構成画像のファイル名  
**outputfile** : ヒストグラムの出力ファイル名

使用例 :  
d:\data\010226e> tif2hst rec0050.tif rec0050.hst

意味 :  
rec0050.tif の画素値と変換係数から、再構成画像のヒストグラムを求める。

備考 :  
このソフトでは、rec????.tif に埋め込まれている情報(吸収係数への変換係数)を基に画像を取り扱う。したがって、rec2tif.exe あるいは tif\_h2o.exe によって作成された画像以外では正しい動作はしない。出力ファイルの中身はアスキー形式で、左下のようにになっている。任意のグラフソフトで描画することが可能で、右下の図のようになる。  
出力ファイルは左のカラムから順に、吸収係数( $\text{cm}^{-1}$ )・一番目のカラムが正の場合の画素数・一番目のカラムが負の場合の画素数、となっている。右下のグラフで言うと、X軸が一番目のカラムで、黒い曲線が二番目のカラム。赤い曲線が三番目のカラムである。

```
***rec0050.hst の中身***  
0.010000    15510  14990  
0.020000    15468  14883  
0.030000    15482  14551  
0.040000    15428  14240  
(以下省略)  
*****
```



## 1) tif\_conv.exe

機能 :  
tiff 画像を 16bit もしくは 8bit に変換する

記述 :  
%tif\_conv input.tif output.tif destbit

引数 :  
input.tif : 入力ファイル名  
output.tif : 出力ファイル名  
destbit : 出力ファイルの bit 数(8 or 16)

使用例 :  
1. d:¥data¥010226e> tif\_conv test12.tif test16.tif 16  
2. d:¥data¥010226e> tif\_conv test12.tif test8.tif 8  
3. d:¥data¥010226e> tif\_conv test16.tif test8.tif 8

意味 :  
1. test12.tif(12bit)を test16.tif(16bit)に変換。  
2. test12.tif(12bit)を test8.tif(8bit)に変換。  
3. test16.tif(16bit)を test8.tif(8bit)に変換。強制的にスケールングされる。

備考 :  
16bit から 8bit あるいは、12bit から 8bit に変換するときは、強制的にスケールングされるが、8bit から 16bit や 12bit から 16bit への変換ではスケールングされずに、値がそのまま代入されるだけになる。  
出力ファイルは、8bit もしくは 16bit のみ選択可能。

### m) tif\_inf.exe

機能 : 再構成画像あるいはシノグラムの tif 画像の情報を表示する

記述 :  
%tif\_inf input

引数 :  
**input** : 情報を表示したい画像ファイルの名前

使用例 :  
d:¥data¥010226e> tif\_int **rec0300.tif**  
d:¥data¥010226e> tif\_int **sino0300.tif**

意味 :  
rec0300.tif あるいは sino0300.tif のファイル情報を表示する。

備考 :  
再構成画像(CT 像)の場合、表示される情報は次のとおり。

1. ビット数
2. 横ピクセル数
3. 縦ピクセル数
4. 投影数
5. ピクセルサイズ
6. 変換係数(div)
7. 変換係数(base)
8. 回転軸の位置
9. フィルタ関数

シノグラムの場合、表示される情報は次のとおり

1. ビット数
2. 横ピクセル数
3. 縦ピクセル数(=投影数)
4. 投影数
5. 変換係数(div)
6. 変換係数(base)
7. 回転軸の位置(重心演算で推測された値)

## 2 . Windows 用バッチファイル

パッケージには3つのバッチファイルが含まれている。

### a) . mkdark.bat

3 ページの処理番号 0 を実行する “**mkdark.bat**” には

```
img_ave d01.img d02.img dark.img
```

と一行だけ記述されている。実験時に取得した暗電流データが **d01.img** と **d02.img** の場合、このバッチファイルを生データのあるフォルダの中で実行するだけで、二つのファイルの平均値を得ることができる。

画像再構成には3ページに書かれているすべての手順が必要である(1・2・3)。“**reconst.bat**”はこれを実行する。やや煩雑な記述がなされているが本質的には

```
ct_sino %1 q dark.img  
sino_conv %2  
ct_cbp  
rec2tif 8
```

という記述しかない。一つ目の引数(%1 と書いてある所)は**必須**で、再構成する layer を指定する。二つ目の引数(%2)は回転軸の位置を入力するもので、重心計算がうまくいかないときに手動で回転軸の位置を決めるときに入力するとよい。4行目の **rec2tif** では 8bit TIFF 形式に出力させている。

1 と 3 を実行する “**sinogram.bat**” には

```
ct_sino %1 q dark.img  
sino2tif 8
```

と書いてある。一つ目の引数(%1)は**必須**で、シノグラムを作る layer を指定する。2行目の **sino2tif** で 8bit TIFF ファイルに出力させている。

### 3 . path の通し方(windows)

windows98 では、“c:¥autoexec.bat”をメモ帳などで見ると、次のような記述があるかもしれない。

```
****  
SET PATH=C:¥PROGRA~1¥latex¥bin; "%PATH%"  
****
```

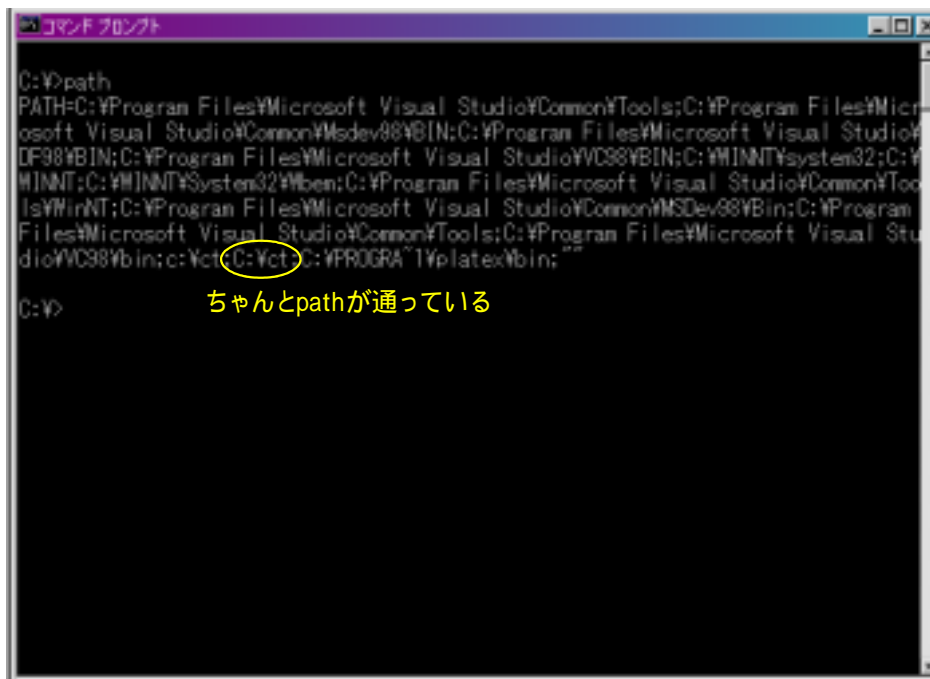
最初の例で示したように、“c:¥ct”にすべてのバッチファイルと実行形式を展開したとすると、これを次のように書き換え、上書き保存したあとにマシンを再起動することにより環境が再設定される。

```
****  
SET PATH=c:¥ct;C:¥PROGRA~1¥latex¥bin; "%PATH%"  
****
```

また、場合によっては“c:¥autoexec.bat”に何も書かれていない事もあるかもしれない。そのときは次の一行を書き、上書き保存したあとマシンを再起動する。

```
SET PATH=c:¥ct; "%PATH%"
```

再起動後に path がちゃんと通っているかを確認するには、DOS 窓で“path”というコマンドを実行すればよい(次の図)。



同じものが2つ通っているように見えるのはご愛嬌・・・

windows 2000 や windowsXP では、コントロールパネル - システム - 環境変数にある path のところに、;c:¥ct を書き足せばよい。その後コマンドプロンプトを起動すれば環境変数は反映されている。確認の方法は上と同じように、コマンドプロンプトで path と入力するだけ。

## 画像再構成の原理

CT 実験でよく用いられている Filtered Back Projection 法について説明する。下の図のような配置において、物体の projection 像は

$$p(x', \theta) = \ln(I_0 / I) = \int \mu(x, y) dy' \quad \dots 1$$

で得られる。ただしこの場合、X 線は単色光であると仮定する。 $x'$  に対して  $p(x', \theta)$  をフーリエ変換すると、

$$\hat{p}(\omega, \theta) = \int p(x', \theta) \cdot \exp(-i\omega x') dx' \quad \dots 2$$

ここで、1 式を代入すると

$$\hat{p}(\omega, \theta) = \iint \mu(x, y) dy' \cdot \exp(-i\omega x') dx' \quad \dots 3$$

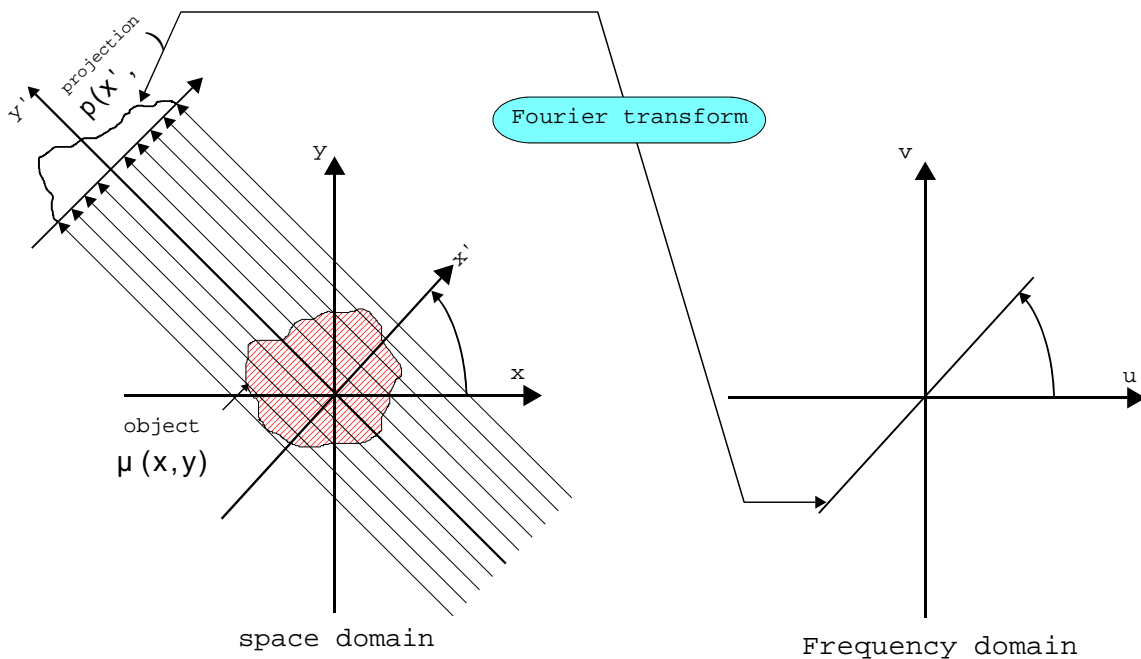
となる。

$$\begin{cases} x' = x \cos \theta + y \sin \theta \\ y' = -x \sin \theta + y \cos \theta \end{cases}$$

だから、

$$\begin{aligned} \hat{p}(\omega, \theta) &= \iint \mu(x, y) \cdot \exp(-i\omega x \cos \theta - i\omega y \sin \theta) dx' dy' \\ &= \iint \mu(x, y) \cdot \exp(-i\omega x \cos \theta - i\omega y \sin \theta) dx dy \quad \dots 4 \end{aligned}$$

となる。ここで、Jacobian,  $|J|=1$  である。



また、

$$\begin{cases} \omega_x = \omega \cos \theta \\ \omega_y = \omega \sin \theta \end{cases}$$

とすると、 $\mu(x, y)$ の2次元フーリエ変換は次のようになる。

$$\hat{\mu}(\omega_x, \omega_y) = \iint \mu(x, y) \cdot \exp(-i\omega_x x - i\omega_y y) dx dy \quad \dots 5$$

つまり  $\hat{p}(\omega, \theta) = \hat{\mu}(\omega_x, \omega_y)$  である。5式の逆フーリエ変換は

$$\begin{aligned} \mu(x, y) &= \frac{1}{(2\pi)^2} \iint \hat{\mu}(\omega_x, \omega_y) \cdot \exp(i\omega_x x + i\omega_y y) d\omega_x d\omega_y \\ &= \frac{1}{(2\pi)^2} \iint \hat{p}(\omega, \theta) \cdot \exp(i\omega_x x + i\omega_y y) d\omega_x d\omega_y \\ &= \frac{1}{(2\pi)^2} \iint \hat{p}(\omega, \theta) \cdot \exp(i\omega x') |\omega| d\omega d\theta \\ &= \frac{1}{2\pi} \int_0^{2\pi} p'(x', \theta) d\theta \end{aligned}$$

ここで、 $p'(x', \theta)$ は

$$p'(x', \theta) = \frac{1}{2\pi} \int \hat{p}(\omega, \theta) \cdot \exp(i\omega x') \cdot g(\omega) d\omega$$

である。式の導出そのままでは、 $g(\omega) = |\omega|$ であるが、 $g(\omega)$ はいわゆる「再構成フィルタ関数」であり、画像再構成のときには適当な $\omega$ 関数に置き換えることが多い。

以上をまとめると、Filtered Back Projection法の基本式は3つであり、それは以下のように表される。

$$\hat{p}(\omega, \theta) = \int p(x', \theta) \cdot \exp(-i\omega x') dx'$$

$$p'(x', \theta) = \int \hat{p}(\omega, \theta) \cdot \exp(i\omega x') \cdot g(\omega) d\omega$$

$$\mu(x, y) = \int_0^{\pi} p'(x \cos \theta + y \sin \theta, \theta) d\theta$$