

核医学機器工学概論 2

断層画像CT(Computed Tomography) を得る方法

1. フィルタ重畠逆投影法

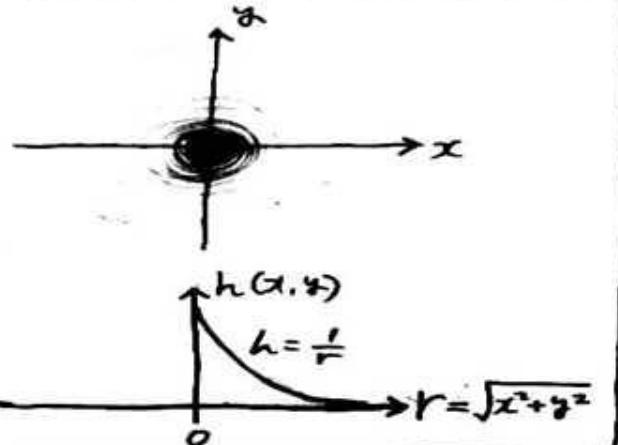
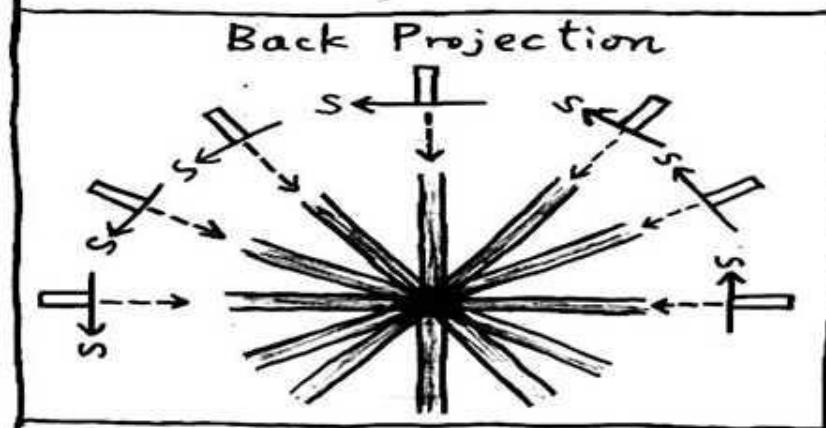
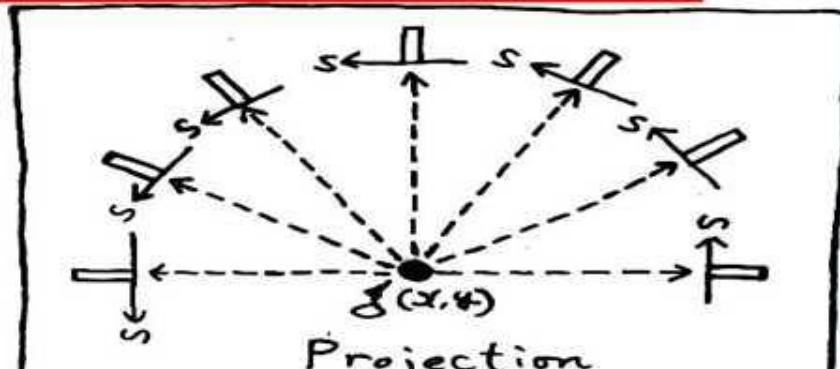
FBP (Filtered Back Projection)

2. 逐次近似再構成法 Iterative Reconstruction

MLEM (Maximum Likelihood Expectation Maximization)

OSEM (Ordered Subsets Expectation Maximization)

⑤ Back Projection



× 平面上の 1 点 $g(x, y)$ の
透視データ $P_\theta(s)$ (Projection)
を $\theta = 0 \sim \pi$ の範囲で求めよ。

次に $P_\theta(s)$ を重ね合わせて
画像の再構成を行なう
(Back Projection)

得られる画像 $h(s, \theta)$ は

$$h(s, \theta) = \sum P_\theta(s) \Delta\theta \\ = \int_0^\pi P_\theta(s) d\theta$$

$h(s, \theta)$ は中心部ほど重ね
合わせ回数が多くなり、中心から
遠ざかるほど少なくなつて、
元の点の像には戻らず、
点の存在した位置からの
距離に反比例した
濃度分布の像をつくる。

重ね合わせの
点像分布関数 $h(x, y)$ は

$$h(x, y) = \frac{1}{r} = \frac{1}{\sqrt{x^2 + y^2}}$$

プログラム PSF.exe の実行。フォルダ PSF 内 の PSF.exe をダブルクリック。

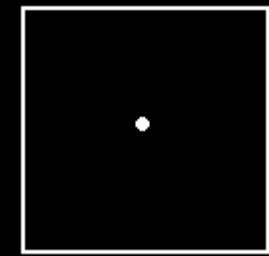
```
Point Spread Function  
PSF BackProjection  
  
max count = 10.000000  
min count = 0.000000  
  
Select Reconstruction method  
1: Simple BackProjection  
2: Filtered BackProjection  
2  
  
Load Real space Filter  
  
Real space filter =  
C:\Users\Katooh\Desktop\医用画像機器工学実  
OK ? (yes; enter, no; n )  
  
Disp Filtered Pth * Filter  
  
maxp count = 76308.327148  
minp count = -8226.914328  
  
Disp FBP Process?  
(yes:enter, no:n)  
  
HU Center = |
```

このテキストウィンドウ内を
クリックしてから 1を入力して
Simple back projectioを実行。

次にプログラム PSF.exe を
終了、再度実行。
2を入力して
Filtered backprojectionを実行。

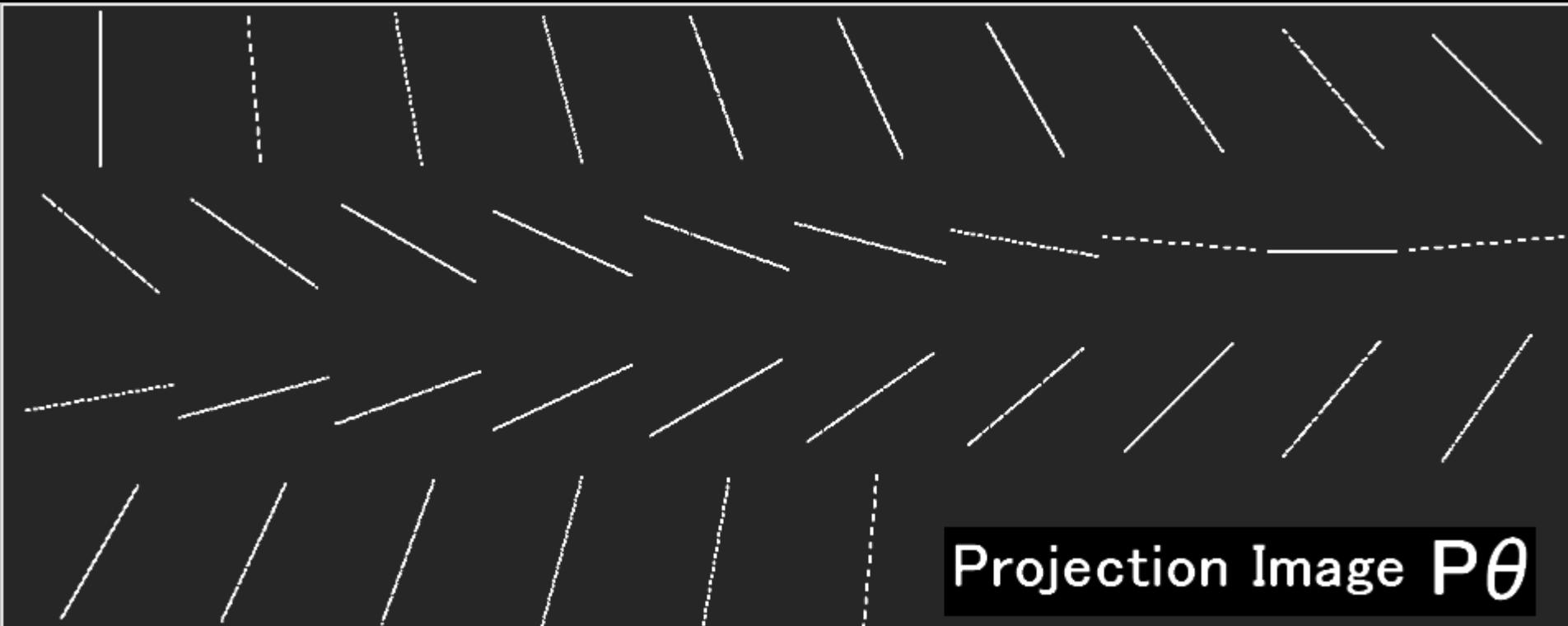
選択する再構成フィルタは、
(real space filter は)
フォルダ PSF 内 にある
RealRAMP256.txt を選択。

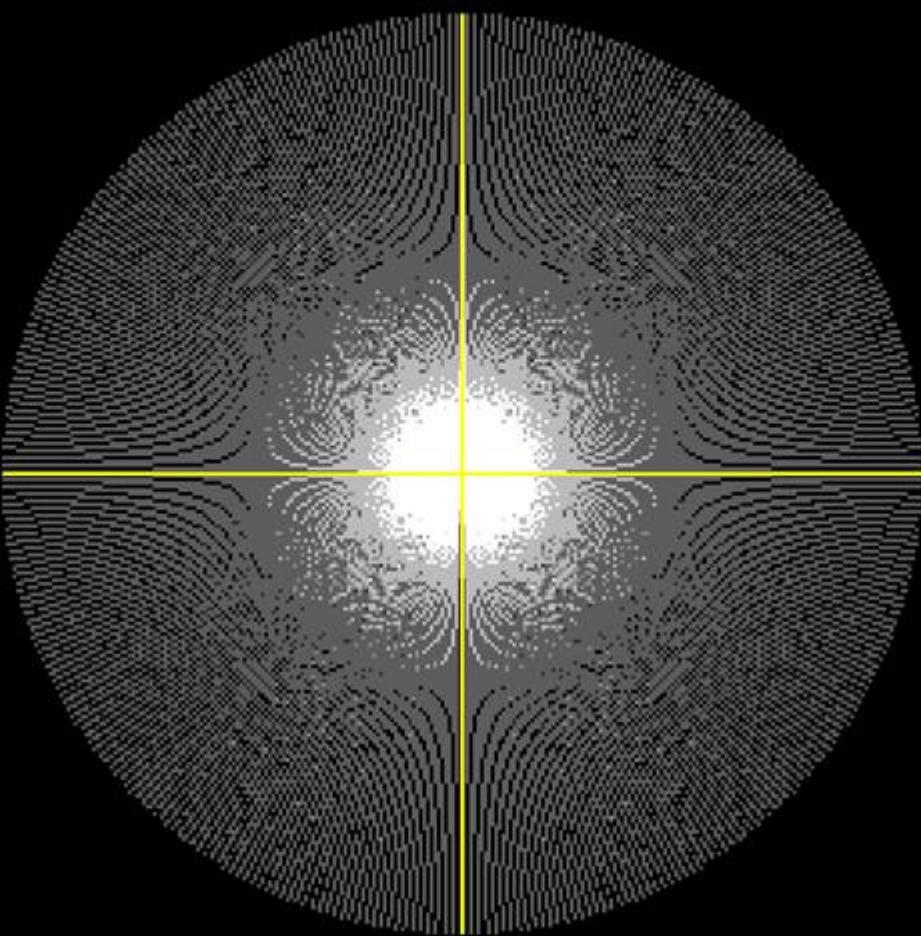
画像中心の1画素だけに画素値を与え、
他を0に設定した2次元画像を作成。



その像を180度方向から横から透視したと想定した
像 $P\theta$ を作成(1度ごと 合計180枚の線状画像)。

(スライドでは5度ごとの画像を表示)

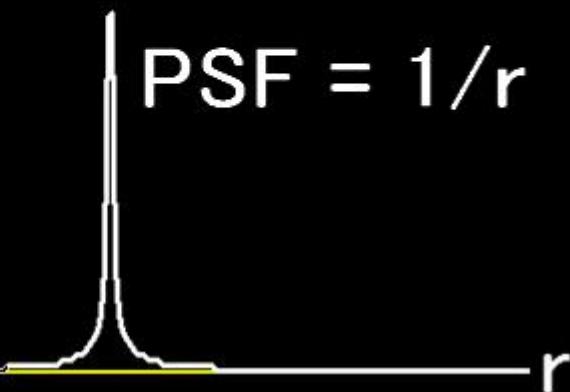




180枚の線状画像 $P\theta$ を重ね合わせると、広がりをもつ分布が得られる。

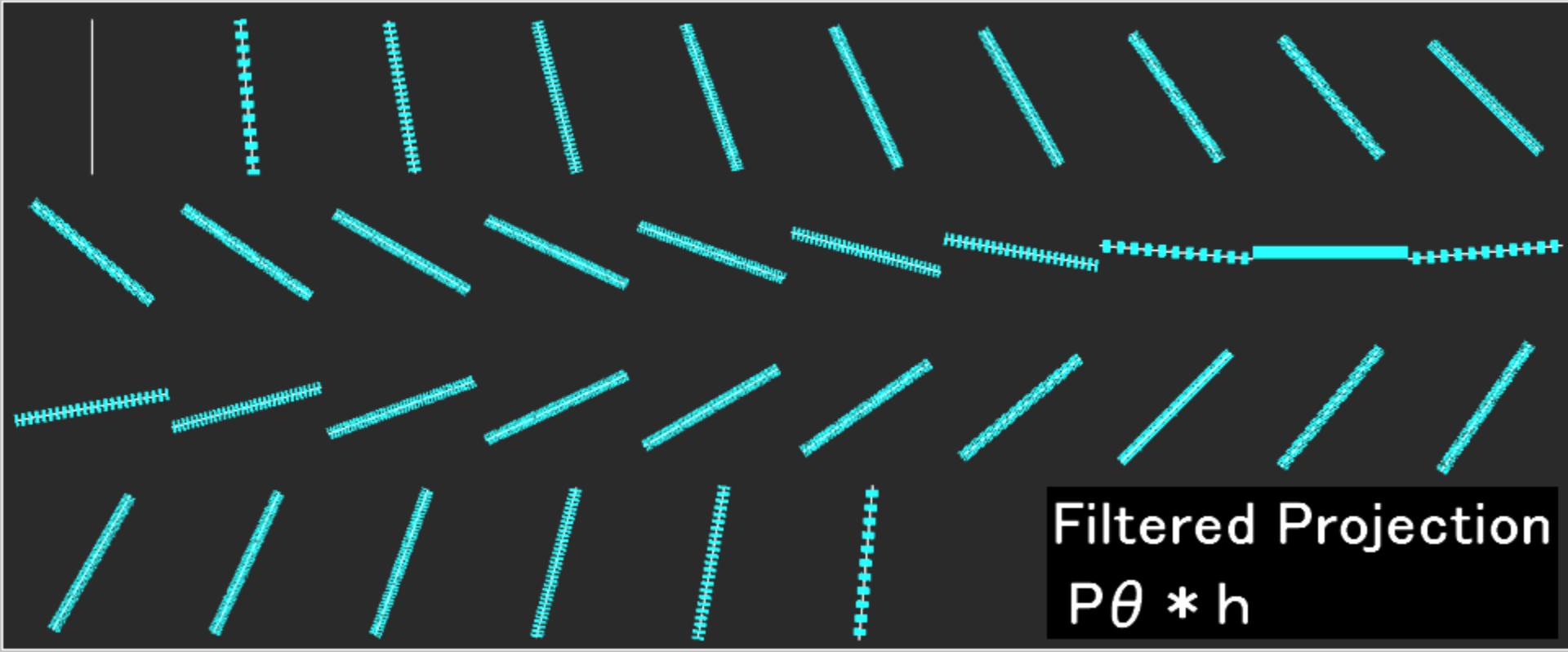
点広がり関数 PSF
(Point Spread Function)

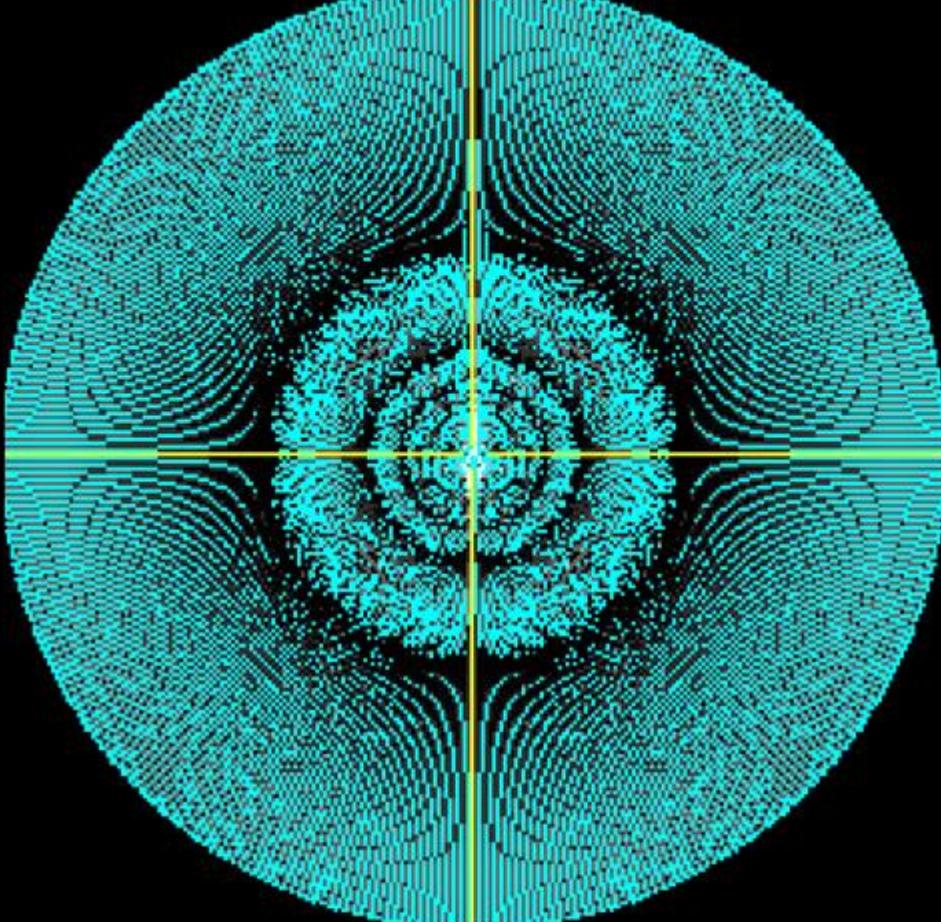
$$\text{PSF} = 1/r$$



180枚の線状画像 $P\theta$ に、
実空間 Ramp (RL) filter h を畳み込む ($P\theta * h$)。

(青く表示された画素はマイナスの値)





180枚の線状画像に
RL filter h を畳み込み
 $P\theta * h$ を作成し、
これらを重ね合わせると、
広がりが消失し、
1点の分布に戻る。
(青はマイナスの画素値)

$PSF * h(r)$ は 点に戻る

Filtered PSF

$1/r * h(r)$

r

回転中心からの距離 r に反比例した濃度に補正するフィルタ

$1/r$ を正確な断層像 g に畳み込んだ像が I である。

式で表現すると $I = g * (1/r)$ となる。

I 、 g 、 $1/r$ のフーリエ変換を L 、 G 、 $F(1/r)$ と表現すると、畳み込みの定理より

$L = G \cdot F(1/r)$ となる。

2次元フーリエ変換の公式の極座標表現を用いると、

(f_r は周波数空間上の原点からの距離)

$F(1/r) =$

$$\iint (1/r) \exp(-j(2\pi r f_r)) r dr d\theta = 1/f_r$$

これより $L = G / f_r$ なので $G = L \cdot f_r$

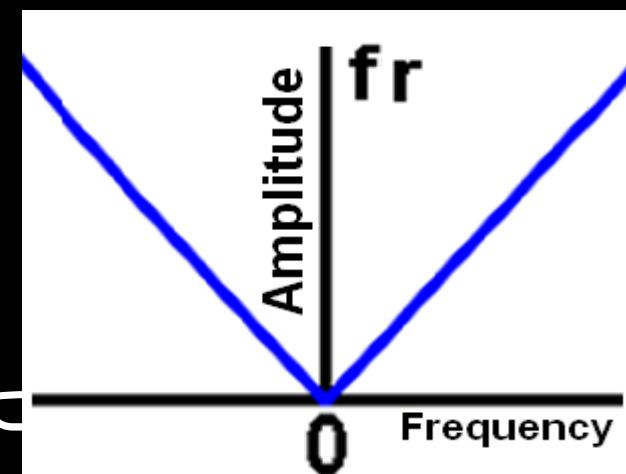
$G = L \cdot f_r$ の意味は、2次元周波数空間上で、
単純重ね合わせ画像をフーリエ変換した2次元データ L に、
フィルタ関数 f_r (f_r は周波数空間上の原点からの距離)
をかけると、正しい再構成画像をフーリエ変換したデータ G
になる。

$G = L \cdot f_r$ に、畳み込みの定理 を

用いると、以下のような実空間での計算に
変換できる。

この式を 逆フーリエ変換すると、

$$g = L * h \quad (h \text{ は フィルタ } f_r \text{ の 逆フーリエ変換 })$$



この式に、 $I = \int P\theta d\theta$ を代入すると、

$$g = \int P\theta d\theta * h$$

$$g = \int (P\theta * h) d\theta \quad (h \text{は } \theta \text{と独立した値なので交換可})$$

$$g = \int \overline{P\theta} d\theta \quad (\overline{P\theta} = P\theta * h)$$

FBPの式

$P\theta$ に 実空間フィルタ h ($= fr$ の逆フーリエ変換)を畳み込めば、重ね合せると正確な断層像 g になる2次元透視画像 $\overline{P\theta}$ を算出できる。これを **Filtered Back Projection (FBP)** という。

周波数空間での実際の計算においては、フィルタ H ($= fr$)は常に正の値であり(絶対値)、

さらにサンプリング定理より、ナイキスト周波数以上の成分を削除する必要があるので、

周波数空間での再構成フィルタ H は、

$$H = |f_r| \quad (f_r \text{ がナイキスト周波数未満の場合})$$

$$H = 0 \quad (f_r \text{ がナイキスト周波数以上の場合})$$

となる。これを Ramp フィルタという。

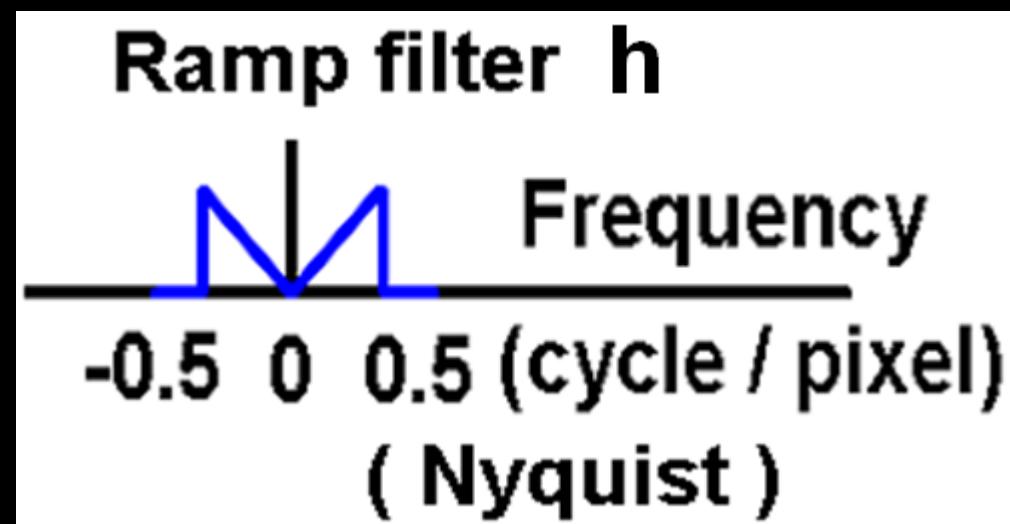
Ramp フィルタを逆フーリエ変換して

実空間 Ramp フィルタ h にしてから、

実空間で $P\theta$ に h を畳み込む。

$$\underline{P\theta} = P\theta * h$$

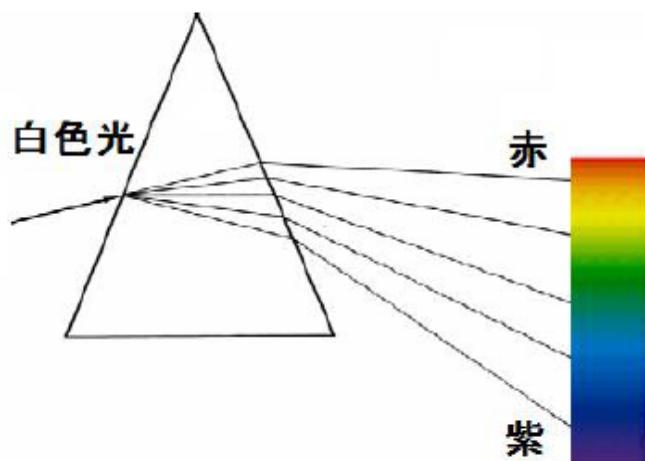
(* は畳み込み演算)



$1/r$ のフーリエ変換が $1/fr$ になる理由

2次元フーリエ変換の公式の極座標表現を用いると

(r と θ は実空間の原点からの距離と偏角、 fr は周波数空間の原点からの距離)
関数 $g(r)$ のフーリエ変換は $\iint g(r) \exp(-j \cdot 2\pi r fr) r dr d\theta$ (ヤコビアンに注意)
 $g(r) = 1/r$ とすると $\iint (1/r) \exp(-j \cdot 2\pi r fr) r dr d\theta = \iint \exp(-j \cdot 2\pi r fr) dr d\theta = 1/fr$

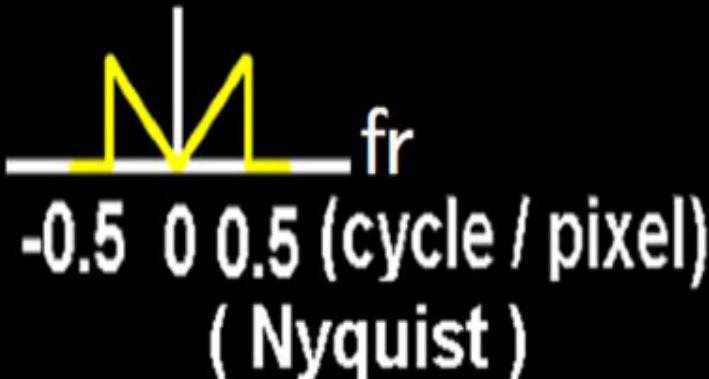


周波数空間を理解することは難しい印象を持つが、周波数空間や周波数分布を表現しているものは一般にも結構ある。

たとえば楽譜は、音楽の周波数分布、曲のフーリエ変換とも解釈できる。五線譜の下側の音符は低音、低周波数成分を表し、上側の音符は高音、高周波成分を示している。

虹は白色光に含まれる色の周波数分布を示している。雨粒中で屈折しやすい高周波の紫色成分から、屈折しにくい低周波の赤色成分まで、周波数の順序で色が並んでいる。

Ramp フィルタ

周波数空間のRampフィルタ H 

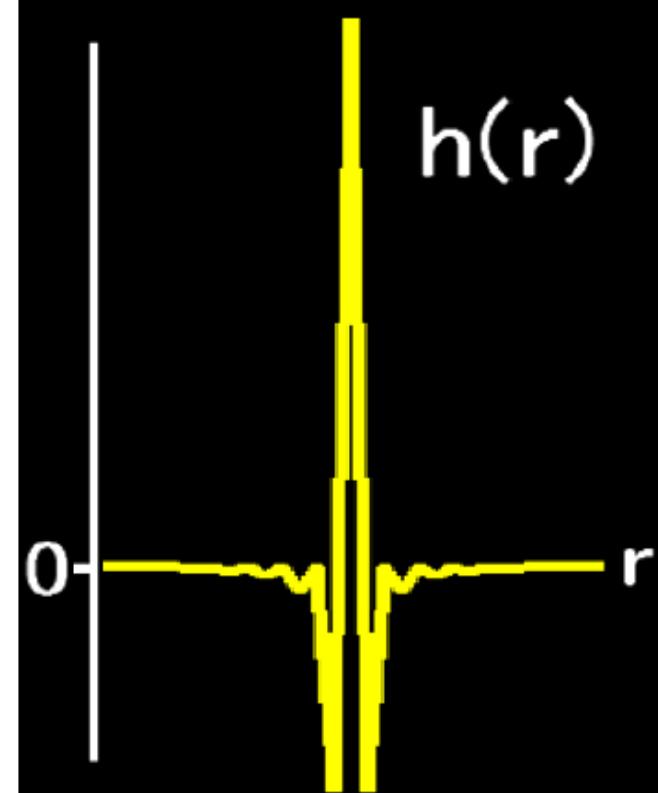
周波数空間でのフィルタ fr とは、どのような形をしているのか。1次元周波数空間の縦軸の周波数成分が fr の値を持つフィルタであるが、横軸が周波数軸すなわち fr である。つまり周波数空間で、正比例のような形状になる。

周波数にはマイナスの値は無いが、高速フーリエ変換(FFT)のアルゴリズムの都合で見かけ上、マイナスの周波数データが存在する。縦軸の周波数成分にはマイナスは無いので、周波数がマイナス側の成分は絶対値となり、原点を中心と左右対称な絶対値 $|fr|$ の関数として表現される。

さらにサンプリング定理によって、ナイキスト周波数 ($0.5 \text{ cycle} / \text{pixel}$) 以上の高周波成分はノイズなので除去する。

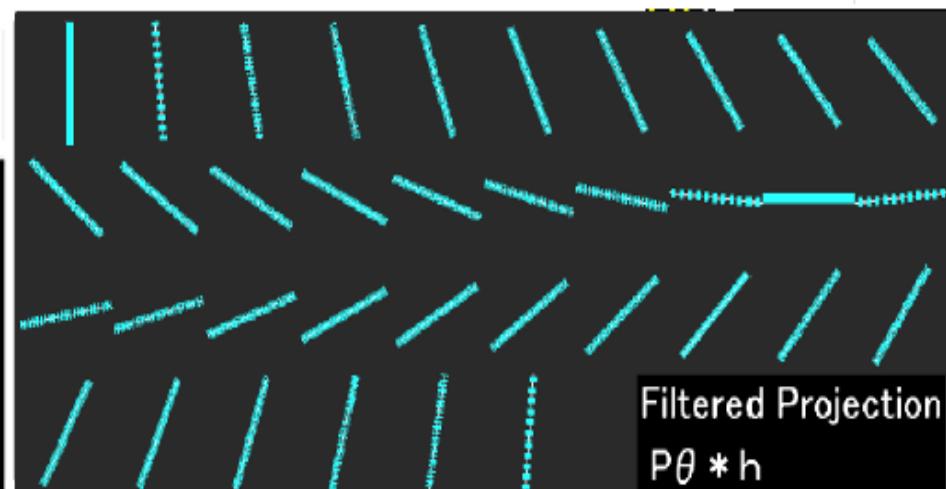
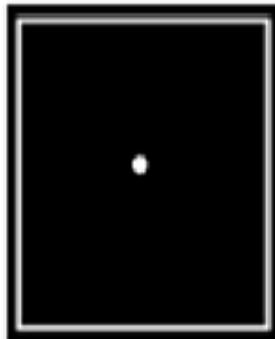
よって、このフィルタは上図のようになり、これをRampフィルタという。

実空間 Ramp フィルタ h

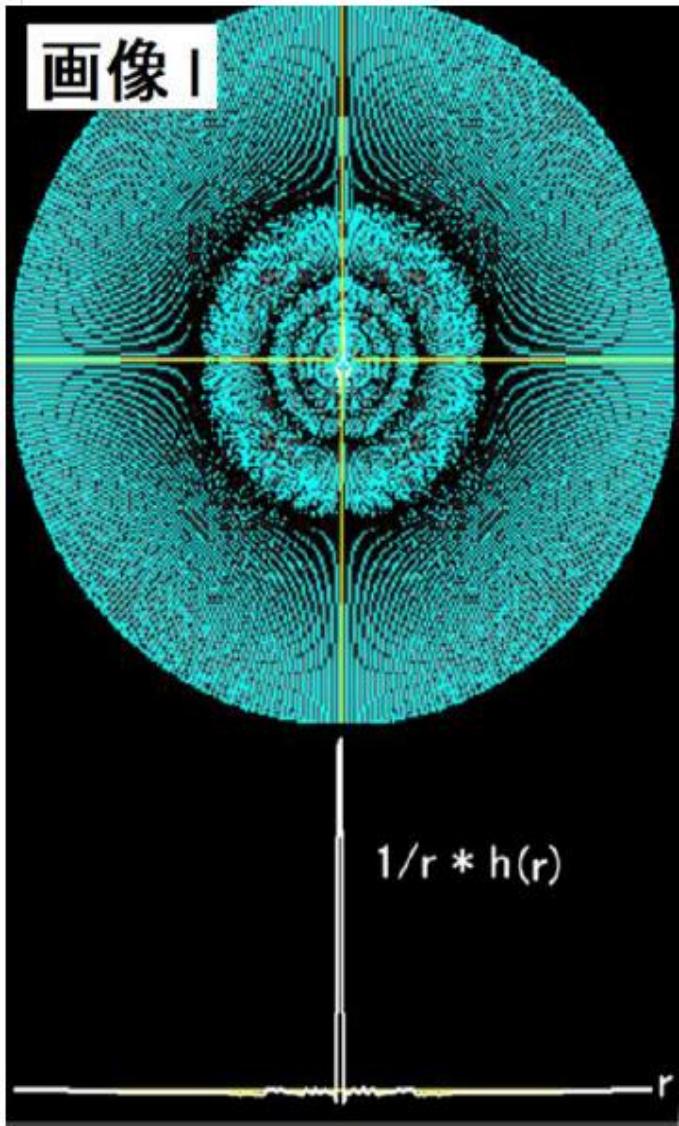


周波数空間の Ramp フィルタ H を表現する。
 H を 1 次元逆フーリエ変換すると、実空間での
Ramp フィルタ h が左図のように算出される。

画像 g



画像 I



画像中心の 1 画素だけ値が 1 で、他は全て 0 の 256x256 画素の画像 g を 1 度毎に 180 度方向から透過投影した投影像 $P\theta$ を作成する。次に全ての $P\theta$ に Ramp フィルタ h を置込む（式で表すと $P\theta * h$ ）（上図で青色の部位はマイナスの画素値）（全ての $P\theta * h$ の断面が h と同じ形状をしている）。さらに、全ての $P\theta * h$ を重ね合わせて画像 I を算出する。式で表すと $I = \int (P\theta * h) d\theta$

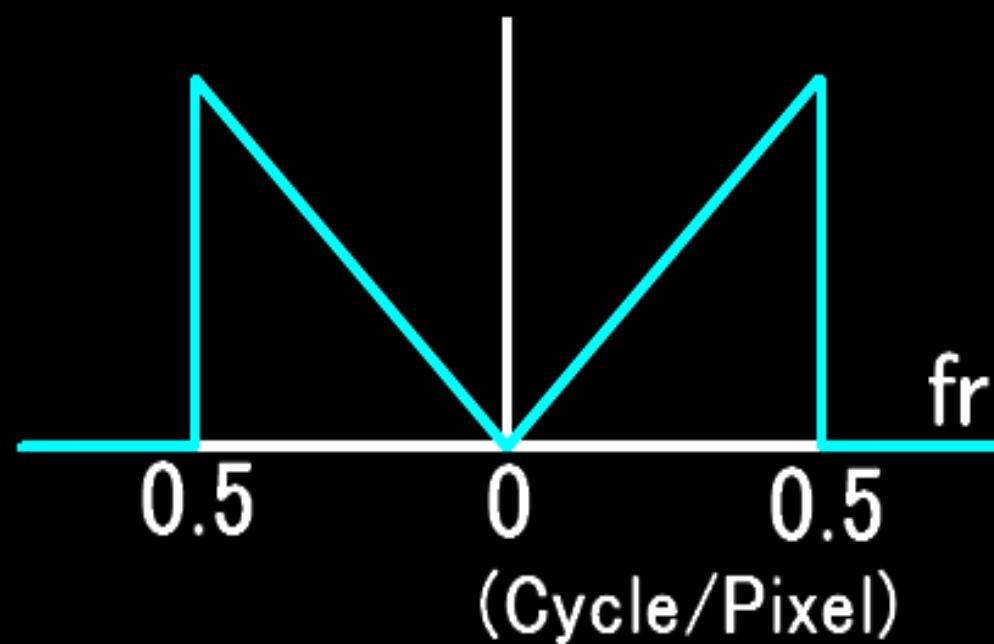
画像 I は中心の 1 画素だけ値が 1 で、周囲は殆ど 0 に近い値を示し、ほぼ画像 g に戻っている。

$P\theta$ の単純重ね合わせ像では、画像中心の 1 画素が最大値を示し、その近傍に中心からの距離 r に反比例する $1/r$ の関数に画素値が広がっていたが、Ramp フィルタ h を置込むと、その広がりが補正されて元の画像が再現される。この Ramp フィルタ h の機能によって断層画像を算出することができる。

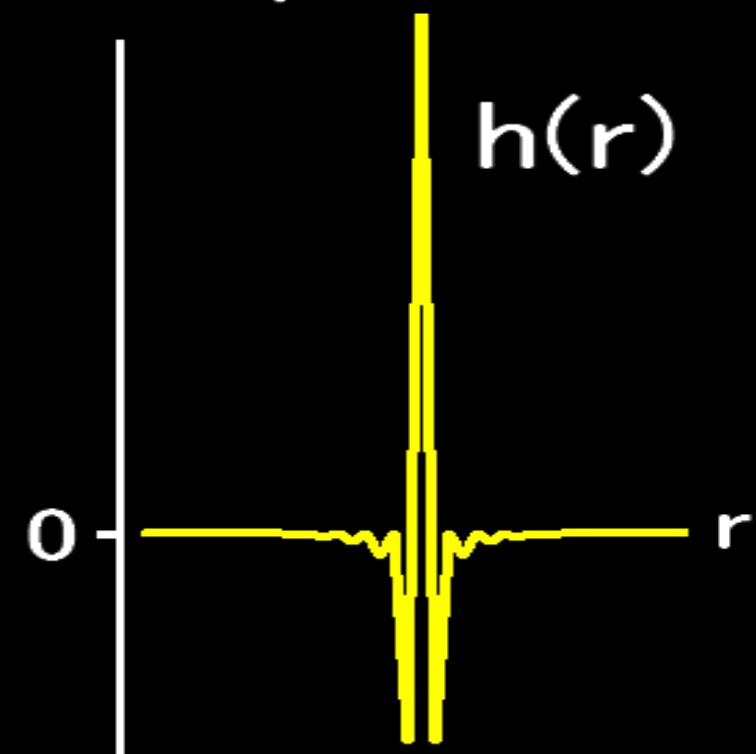
周波数空間 RL フィルタ ($1/f_r$) を
1次元逆フーリエ変換して、
実空間 RLフィルタ $h(r)$ を作成。

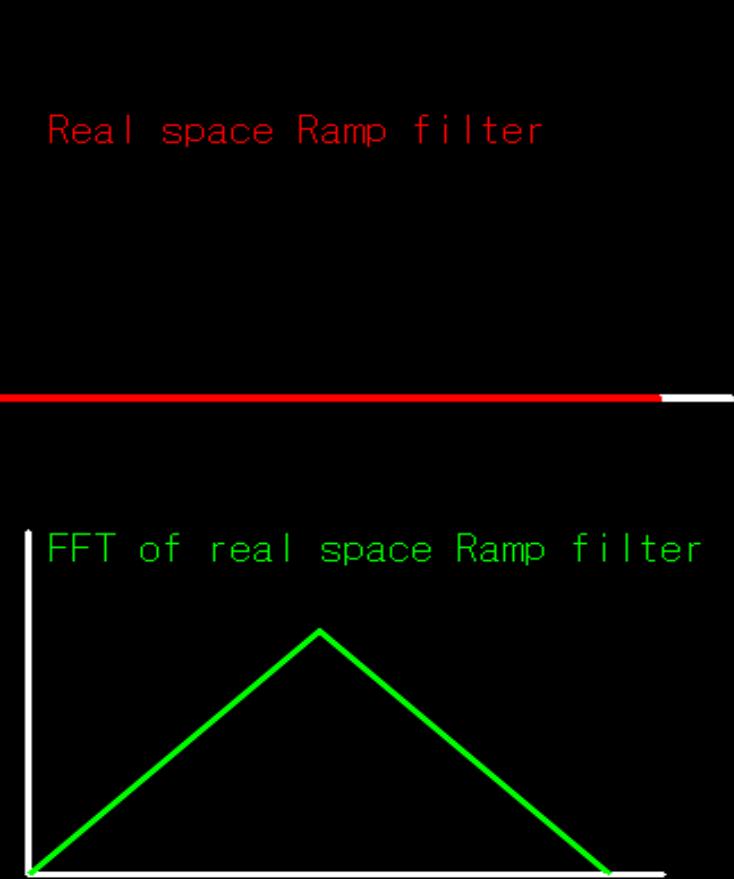
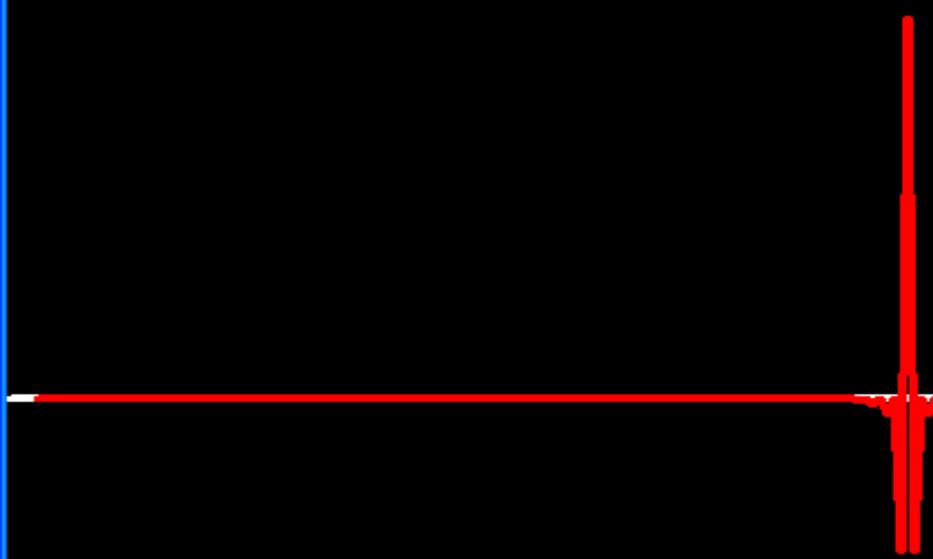
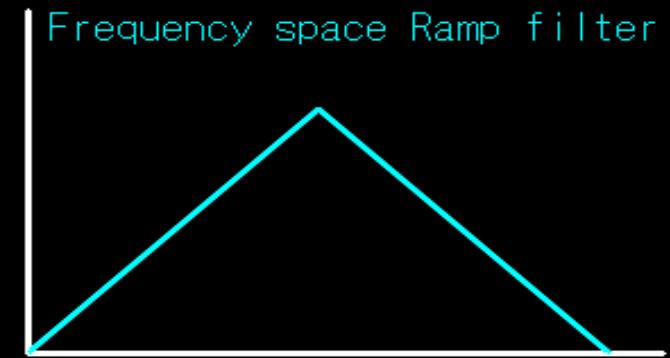
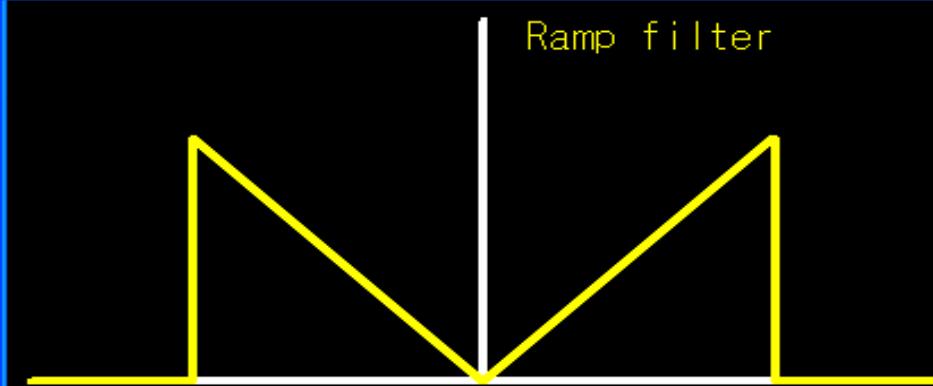
フォルダ RAMP256 内の RAMP256.exeを実行。

Frequency Space RL filter



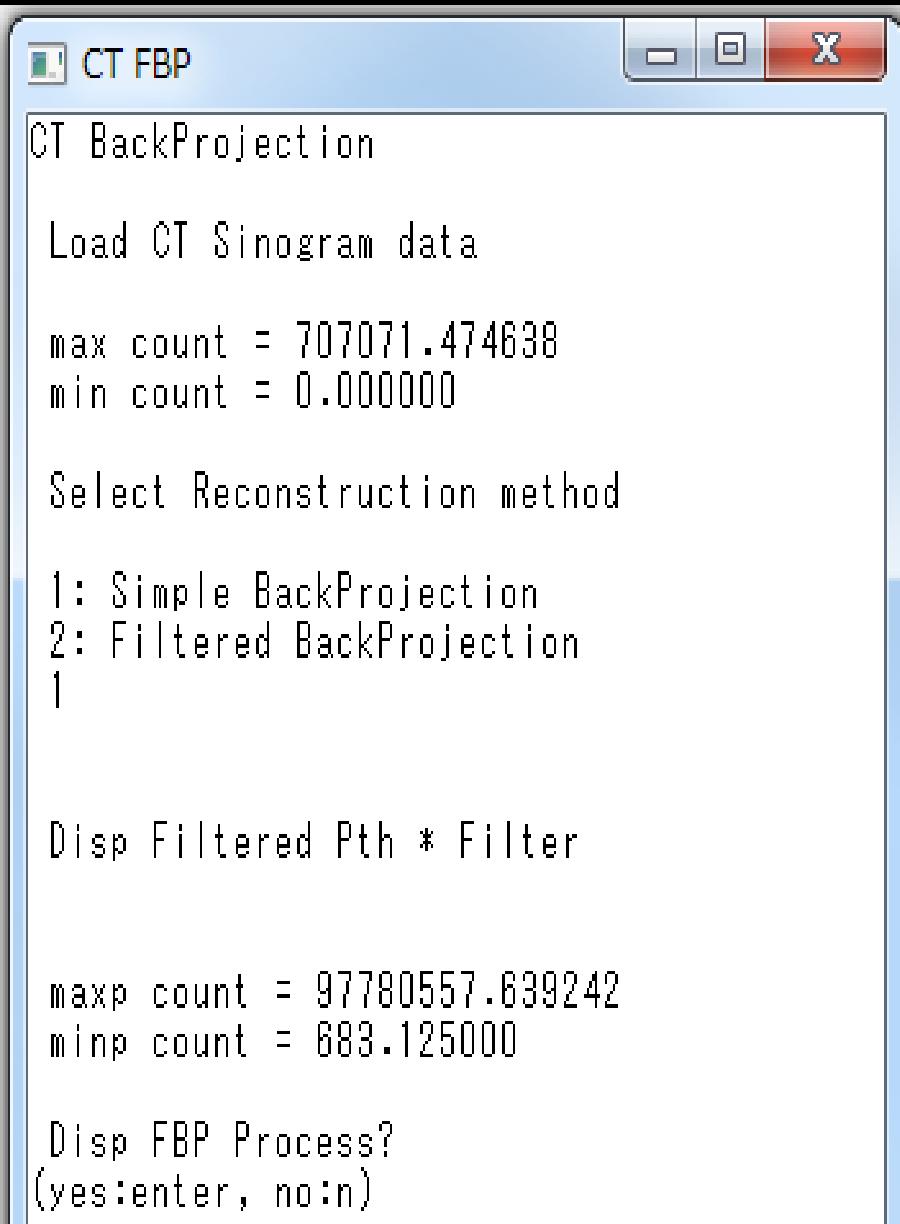
Real Space RL filter





Ramp フィルタの作成
Ramp.c 実行結果

プログラム CTFBP.exe の実行。フォルダ CT 内 の CTFBP.exe をダブルクリック。



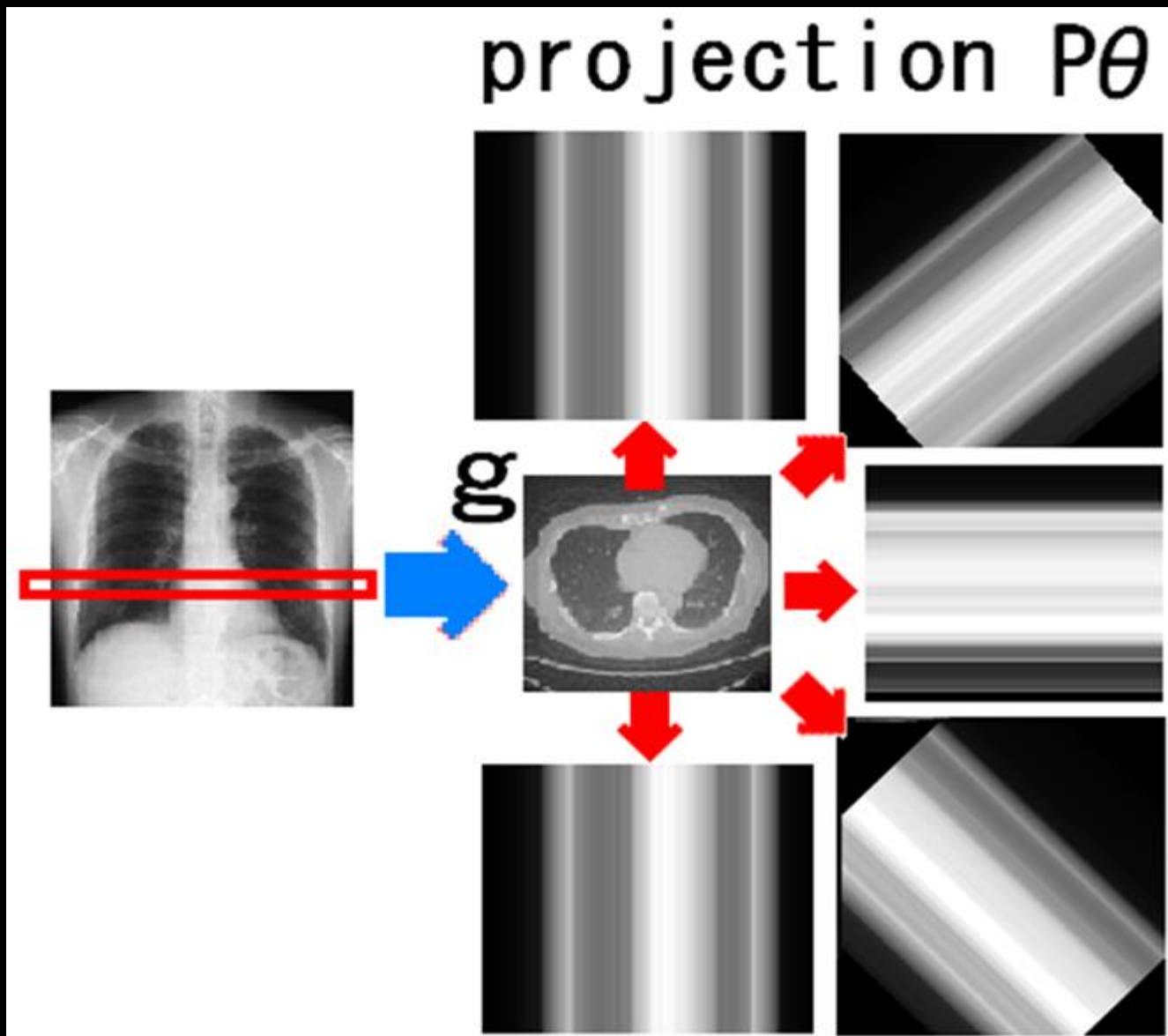
このテキストウィンドウ内を
クリックする。

選択するプロジェクト
データは、フォルダ CT 内 の
CTprojection を選択。

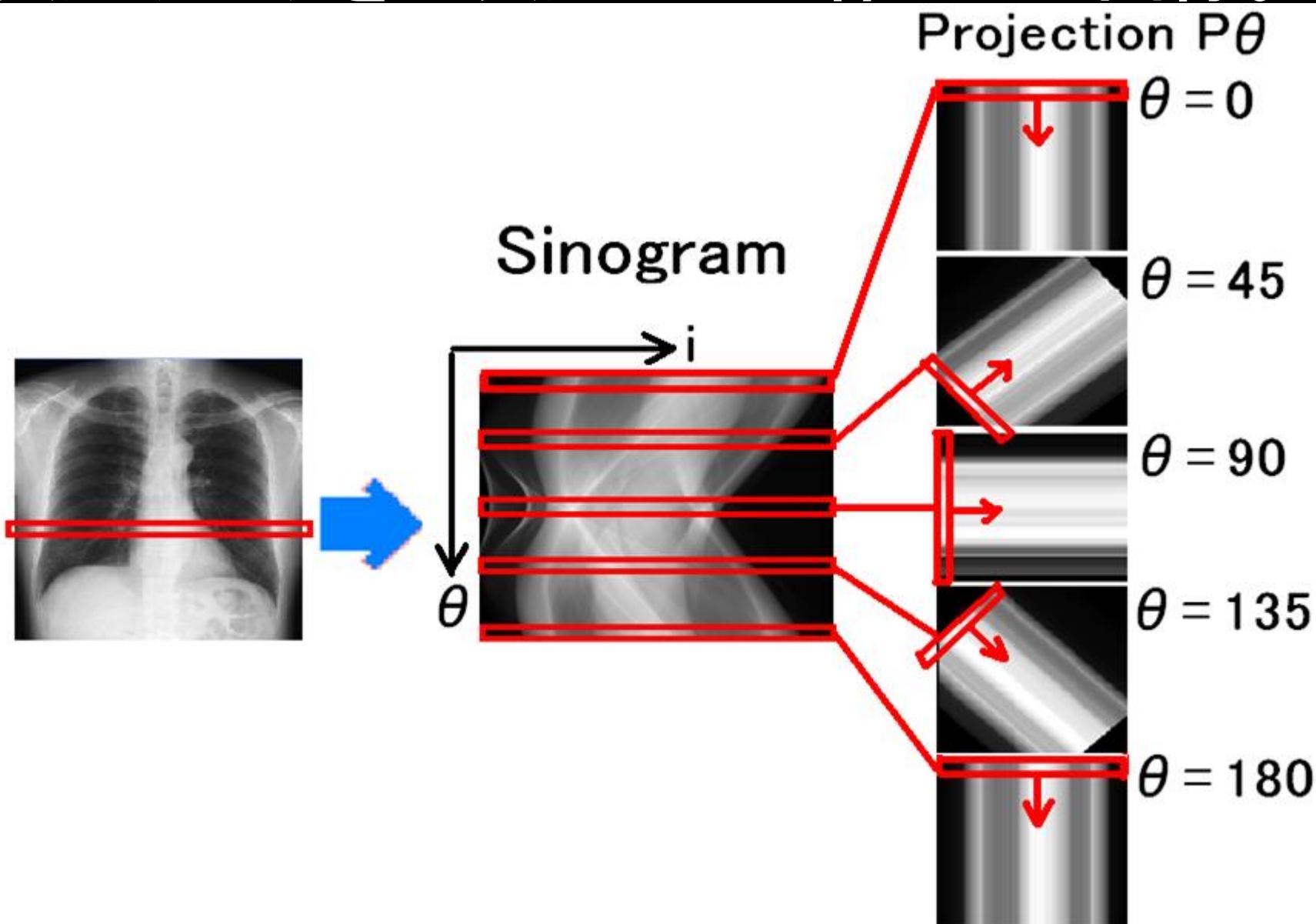
1を入力して
Simple back projectionを実行。

Disp FBP Process? と出たら
Enterキーを押す。
Simple Back projection が
実行される様子を観察。

求めたい正確な断層像 g を算出するために、
多方向の角度 θ から 2次元透視像 $P\theta$ を測定。

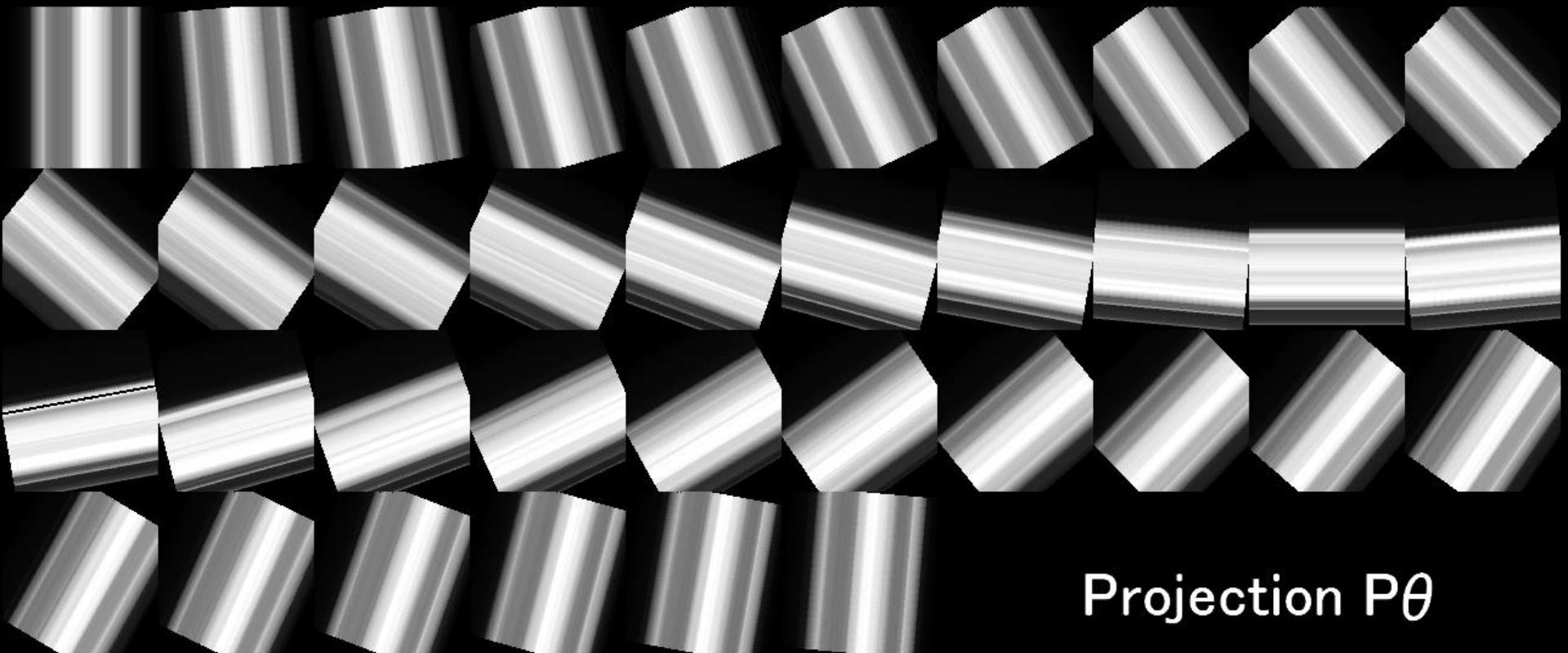


2次元透視像 $P\theta$ は、サイノグラムの各行 θ の
1次元データを 2次元に引き伸ばした画像。



例として胸部の1断面のサイノグラムから
180度方向から横から透視したと想定した像 $P\theta$ を
作成(1度ごと、合計180枚の2次元透視画像)。

(スライドでは5度ごとの画像を表示)



180度方向から透視した像 $P\theta$ を重ね合わせると
ぼけて画像中心の値が持ち上がった像 I を得る。
正確な断層像 g と I の関係式は、

$$I = g * h$$

正確な断層像 g に 点広がり関数 h が 置み込まれ
ぼやけた断層像 I を得る と考える。



単純重ね合わせ再構成法 Simple Back Projection

収集された各々の角度に傾いた2次元透視画像($P\theta$)を全部単純に重ねると再構成画像ができる。
(回転中心近傍の値が盛り上がった不正確な画像。)

スライス j におけるサイノグラムを求める。
サイノグラムの各スライスの 1 次元配列は、
各々の角度から収集されたデータ。

サイノグラムの各スライスの 1 次元配列から、
収集された各々の角度に傾いた 2 次元透視画像 $P\theta$ を作成する。
 $P\theta$ を単純に重ね合わせた画像を I とすると

$$I = \int P\theta d\theta \quad (\text{Simple back projection})$$

I は、回転中心部ほど重ね合せ回数が多くなり、
中心から距離が遠いほどカウントの低い像になる。

プログラム CTFBP.exe の再実行。 フォルダ CT 内 のCTFBP.exe をダブルクリック。



CT FBP

```
CT BackProjection
Load CT Sinogram data
max count = 707071.474638
min count = 0.000000
Select Reconstruction method
1: Simple BackProjection
2: Filtered BackProjection
2

Load Real space Filter
Real space filter =
C:\Users\Katoh\Desktop\医用画像機器工学実
OK ? (yes; enter, no; n )

Disp Filtered Pth * Filter
maxp count = 170934604.546631
minp count = -12698206.878418
Disp FBP Process?
(yes:enter, no:n)
```

このテキストウィンドウ内を
クリック。

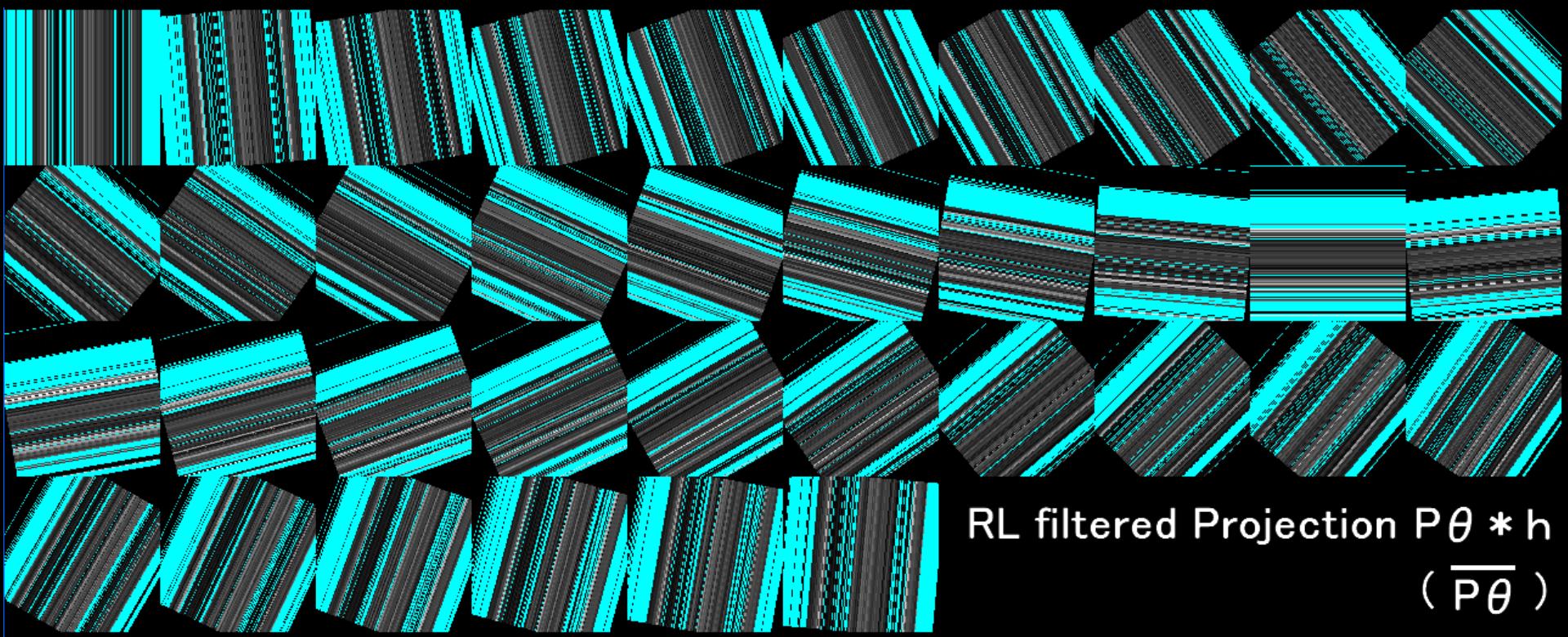
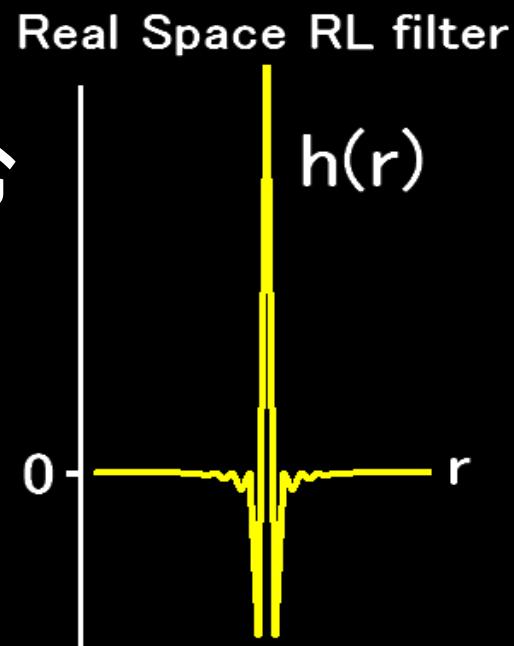
選択するプロジェクト
データは、フォルダ CT 内 の
CTprojection を選択。

2を入力して
Filtered back projectionを実行。
選択するReal space filter は
フォルダ CT 内 の
RealRAMP256.txtを選択。

Disp FBP Process? と出たら
Enterキーを押す。

180枚の2次元透視画像 $P\theta$ に
実空間 Ramp (RL) filter h を畳み込む
($P\theta * h$)。

(青く表示された画素はマイナス値)



RL filtered Projection $P\theta * h$
($\overline{P\theta}$)

$P\theta * h$ を重ね合せると 正確な断層像 g になる。

$$g = \int \overline{P\theta} d\theta \quad (\overline{P\theta} = P\theta * h)$$

FBPの式



畳み込みの定理

データ g をフーリエ変換して、

その周波数空間成分 G に

周波数空間 Ramp フィルタ H をかけて

逆フーリエ変換すると、

実空間で、実空間 Ramp フィルタ h を

g に畳み込みしたデータと同じになる。

($G \times H$ と $g * h$ は等価演算)

スリットを取りはずすと（スリット間隔を無限に狭くすると）
任意の座標 x におけるフィルムの濃度 $l(x)$ は

$$l(x) = \int_{-\infty}^{\infty} g(n) h(x-n) dn \quad \begin{array}{l} \text{Convolution 積分} \\ (\text{たたみ込み 積分}) \end{array}$$

これを $l(x) = g(x) * h(x)$ と表わす。

$h(x)$ を convolution 関数といふ。

$g(x)$ が $h(x)$ のために $|l(x)|$ は ボケてしまつたことを意味する。

たたみ込みの定理

$h(x-n)$ の Fourier 変換を $H(f)$ とする

$$h(x-n) = \int H(f) e^{j(2\pi f(x-n))} df \quad (\text{逆 Fourier 変換})$$

これを $l(x) = \int g(n) h(x-n) dn$ に代入すると

$$\begin{aligned} l(x) &= \int g(n) \left[\int H(f) e^{j(2\pi f x)} e^{-j(2\pi f n)} df \right] dn \\ &= \int \left[\int g(n) e^{-j(2\pi f n)} dn \right] H(f) e^{j(2\pi f x)} df \\ &= \int [G(f)] H(f) e^{j(2\pi f x)} df \end{aligned}$$

$l(x)$ の Fourier 変換を $L(f)$ とする

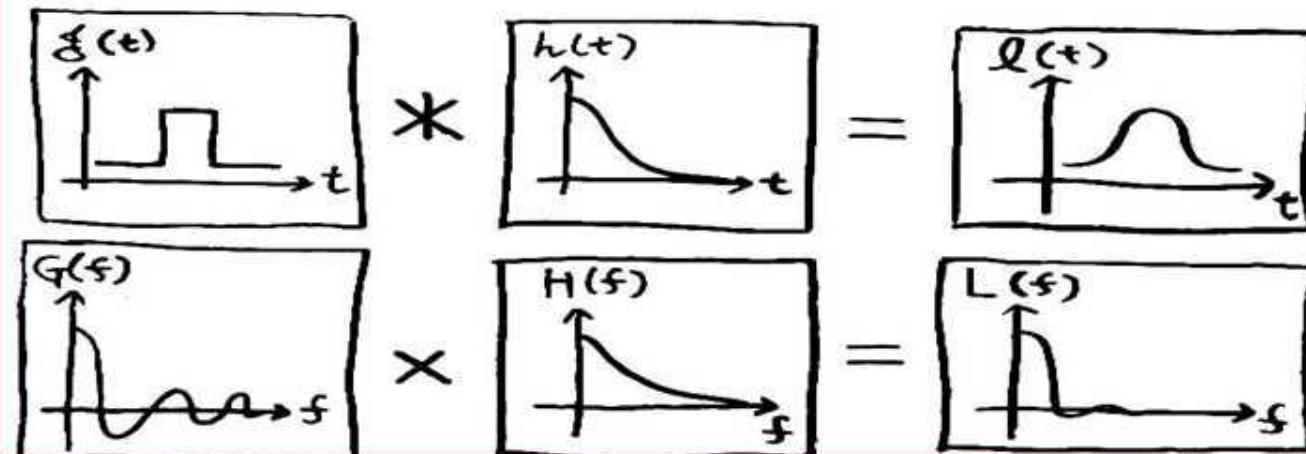
$$l(x) = \int L(f) e^{j(2\pi f x)} df, \text{ よし} \quad L(f) = G(f) H(f)$$

位置(x)や時間(t)の関数 g が 装置の特性 h によって
ボケて l になることは、

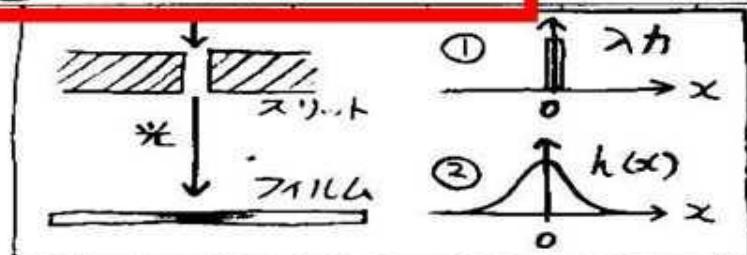
曲線 g の細かい変動(高周波成分)が んによって
減衰して 大ざっぱな 曲線 l になることを示す。

つまり 周波数空間において、 $G(f)$ は $H(f)$ によって
高い周波数の成分だけ減衰させられて
高周波成分の少なくなつた $L(f)$ になる。

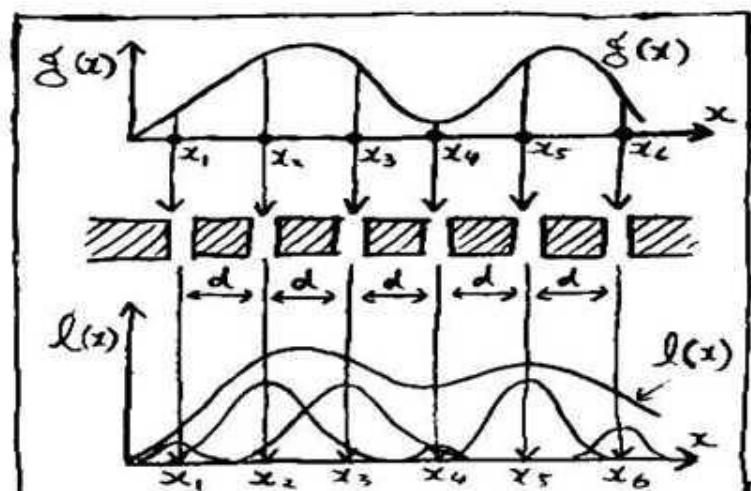
Convolution が周波数空間において 乗算として
示されるることは、 f が大きい値のときに $H(f)$ は 小さい
値をとる filter で、そのため $G(f)$ に $H(f)$ を掛けた
 $L(f)$ は 高周波成分が 小さくなつて、そのために
 l は 大ざっぱな曲線にボケると解釈できる。



(4) Convolution



スリットを通して入射したビーム ① が
フィルム上で ② のような $h(x)$ の
濃度分布の像をつくるとする。

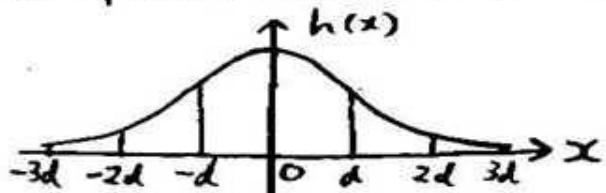


x 軸に沿って明るさが $g(x)$ である
点線状の被写体を、間隔 d の
スリットを通してフィルムに写す。

フィルムの x_4 の位置での濃度は

$$\begin{aligned}l(x_4) = & g(x_4) h(0) \\& + g(x_5) h(-d) + g(x_6) h(-2d) \\& + g(x_3) h(d) + g(x_2) h(2d) \\& + g(x_1) h(3d)\end{aligned}$$

Line spread function (LSF)



任意の座標 x_i における $l(x_i)$ は

$$\begin{aligned}l(x_i) = & g(x_i) h(0) \\& + g(x_{i+1}) h(x_i - x_{i+1}) \\& + g(x_{i+2}) h(x_i - x_{i+2}) + \dots \\& + g(x_{i-1}) h(x_i - x_{i-1}) \\& + g(x_{i-2}) h(x_i - x_{i-2}) + \dots \\& = \sum_{n=-\infty}^{\infty} g(x_n) h(x_i - x_n)\end{aligned}$$

データ数64の1次元データ $g[0] \sim g[63]$ と、
右図に示すようなデータ数32の実空間フィルタ
 $h[0] \sim h[31]$ がある（ $h[0]$ が原点）。

データ g に フィルタ h を畳み込んで
配列 $l[0] \sim l[63]$ に書き込むプログラムを
記述して下さい。



//----- Convolution $h[]$ into $g[]$ -----

```
int i, j;  
double g[64], h[32], l[64], gh;  
  
for( i = 0 ; i <=63 ; i++ ){ // i <64 と書いても同じ  
    gh = 0.0;  
    for( j = 0; j <=31; j++ ){ if( i+j <=63 ) gh += g[ i+j ] * h[ j ] ; }  
    for( j = 1; j <=31; j++ ){ if( i-j >= 0 ) gh += g[ i-j ] * h[ j ] ; }  
    l[i] = gh ;  
}
```

g に h を畳みこんだ結果を I とする。座標 i における I の値 $I[i]$ は、

$$I[i] = g[i]*h[0] + g[i+1]*h[-1] + g[i+2]*h[-2] + g[i+3]*h[-3] + \dots \\ + g[i-1]*h[1] + g[i-2]*h[2] + g[i-3]*h[3] + \dots$$

ここで フィルタ h は偶関数(左右対称)なので、 $h[-j] = h[j]$ を代入し、

$$I[i] = g[i]*h[0] + g[i+1]*h[1] + g[i+2]*h[2] + g[i+3]*h[3] + \dots \\ + g[i-1]*h[1] + g[i-2]*h[2] + g[i-3]*h[3] + \dots$$

これを C 言語で表す。 h の要素数 j は $h[0]$ から $h[31]$ まで。

g の要素数 は $g[0]$ から $g[63]$ までなので、 g の添字を表す $i+j$ は 63 以下、 $i-j$ は 0 以上の場合だけ加算する条件式を入れる。

```
gh = 0.0 ;  
  
for(j = 0; j <= 31; j++){ if( i+j <= 63 ) gh += g[i+j] * h[j] ; }  
for(j = 1; j <= 31; j++){ if( i-j >= 0 ) gh += g[i-j] * h[j] ; }  
I[i] = gh ;
```

これを i が 0 から 63 まで繰り返すと、 $I[0]$ から $I[63]$ が計算される。

プログラム PETFBP.exe の実行。フォルダ PETFBP 内の PETFBP.exe をダブルクリック。



The screenshot shows a window titled "PET FBP". Inside, the application is running a series of commands:

```
PET FBP
Load PET Sinogram data
max count = 472.569855
Disp Projection
select slice = 37
select slice OK ?
Reconstruction slice OK?
(yes: enter, no: n)
Load Real space Filter
Real space filter =
C:\Users\Katoh\Desktop\医用画像機器工学
OK ? (yes: enter, no: n )
Select Reconstruction method
1: Simple BackProjection
2: Filtered BackProjection
2
Disp Filtered Pth * Filter
```

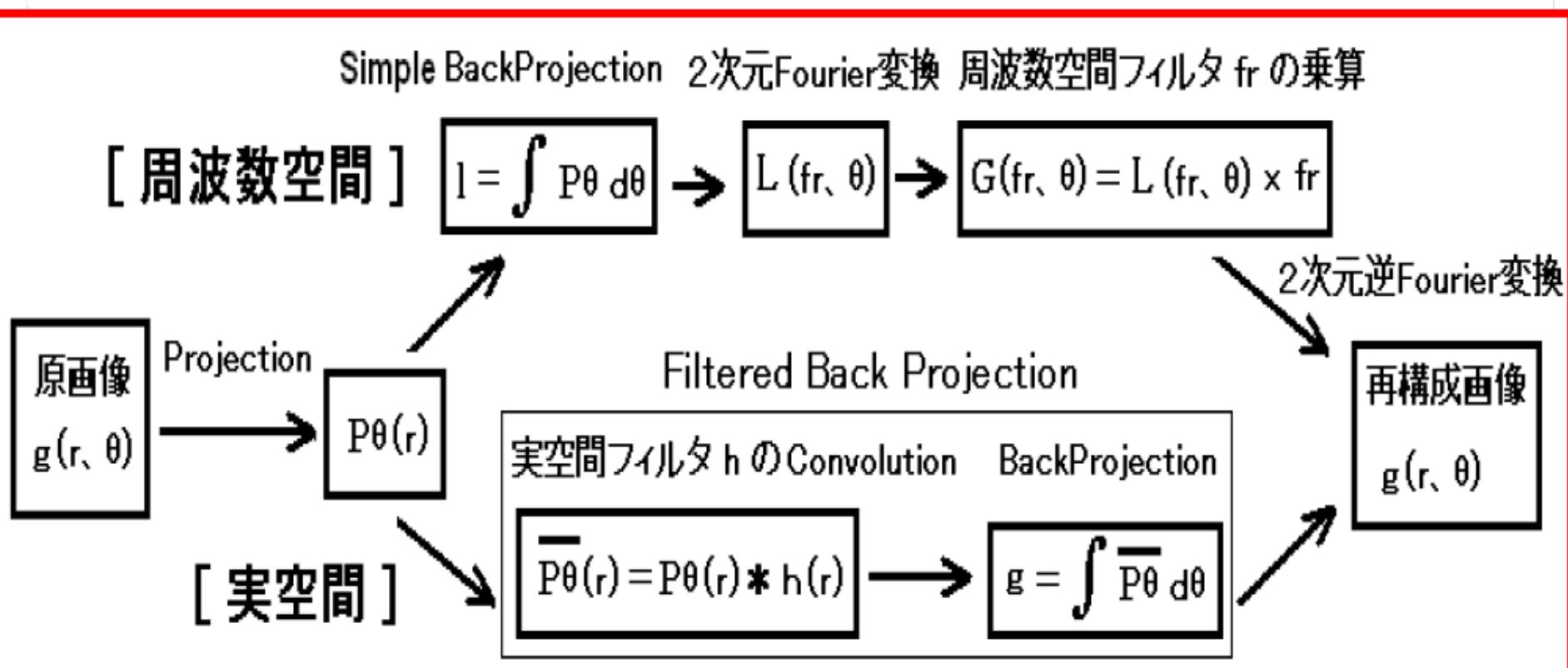
このテキストウィンドウ内を
クリックする。
選択するプロジェクト
データは、フォルダ PETFBP
内の PETsinogram を選択。

Select slice おすすめスライス
は、36、37、38あたり。
2を入力して
Filtered back projectionを実行。
選択フィルタは2種類。
Real|RAMP256. txt
Real|SheppLogan256. txt

投影像 $P\theta$ を単純重ね合わせした画像 I を 2 次元フーリエ変換し、周波数空間でフィルタ fr を乗算したデータを 2 次元逆フーリエ変換すると、正確な断層画像 g が得られる。

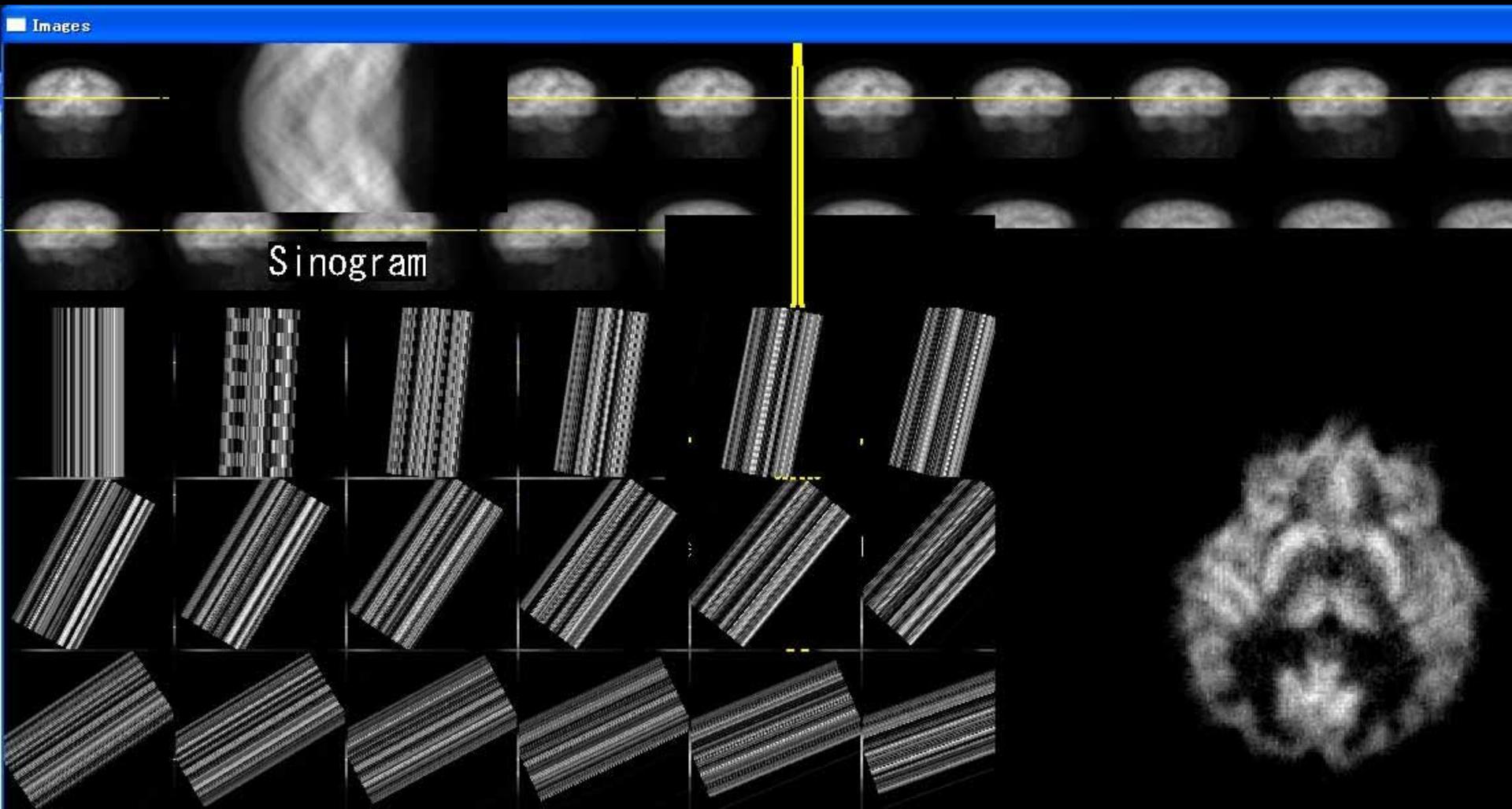
畳込みの定理を使うと、これらの周波数空間での処理が実空間で簡略化される。

投影像 $P\theta$ にフィルタ h (fr の 1 次元逆フーリエ変換) を畳込み、それを重ね合わせると正確な断層画像 g が得られる。



プログラム PETFBP.exe

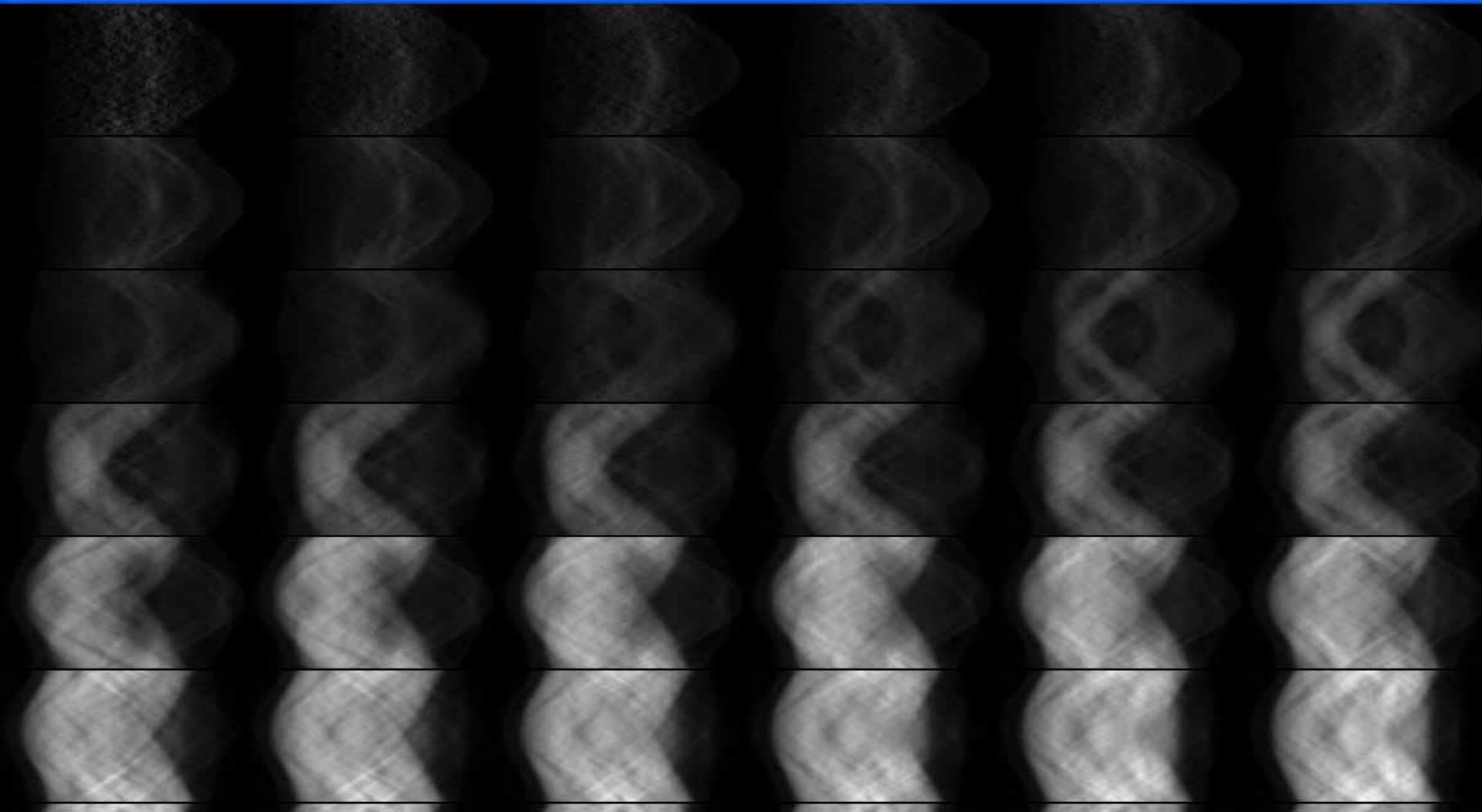
PET画像を Simple BackProjection、または FBP
(Filtered BackProjection) で再構成する。



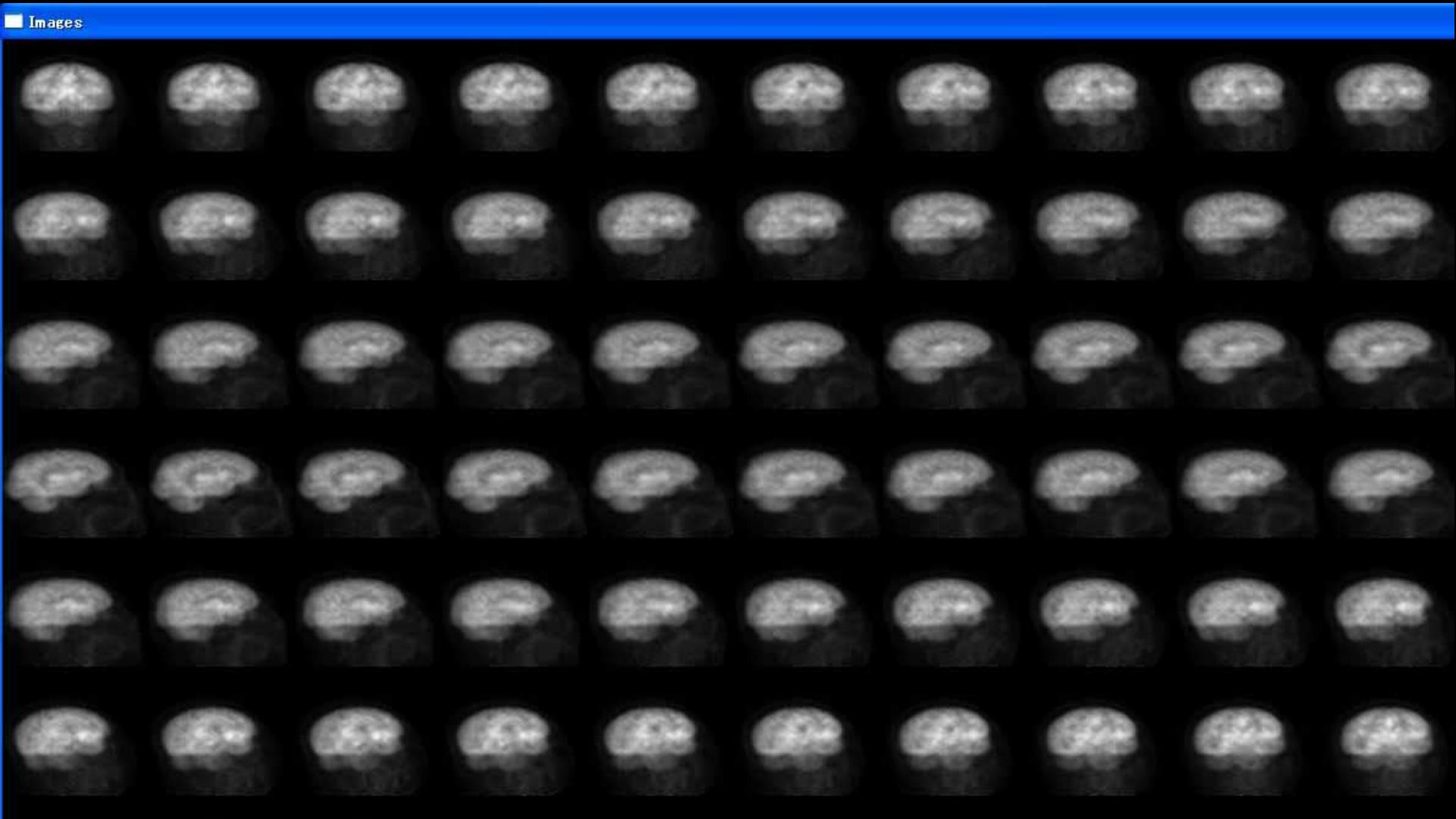
PETsinogram ファイル

PETの収集データは各スライスのサイノグラムが並んでいる 3次元データ。(CTと同じ)。

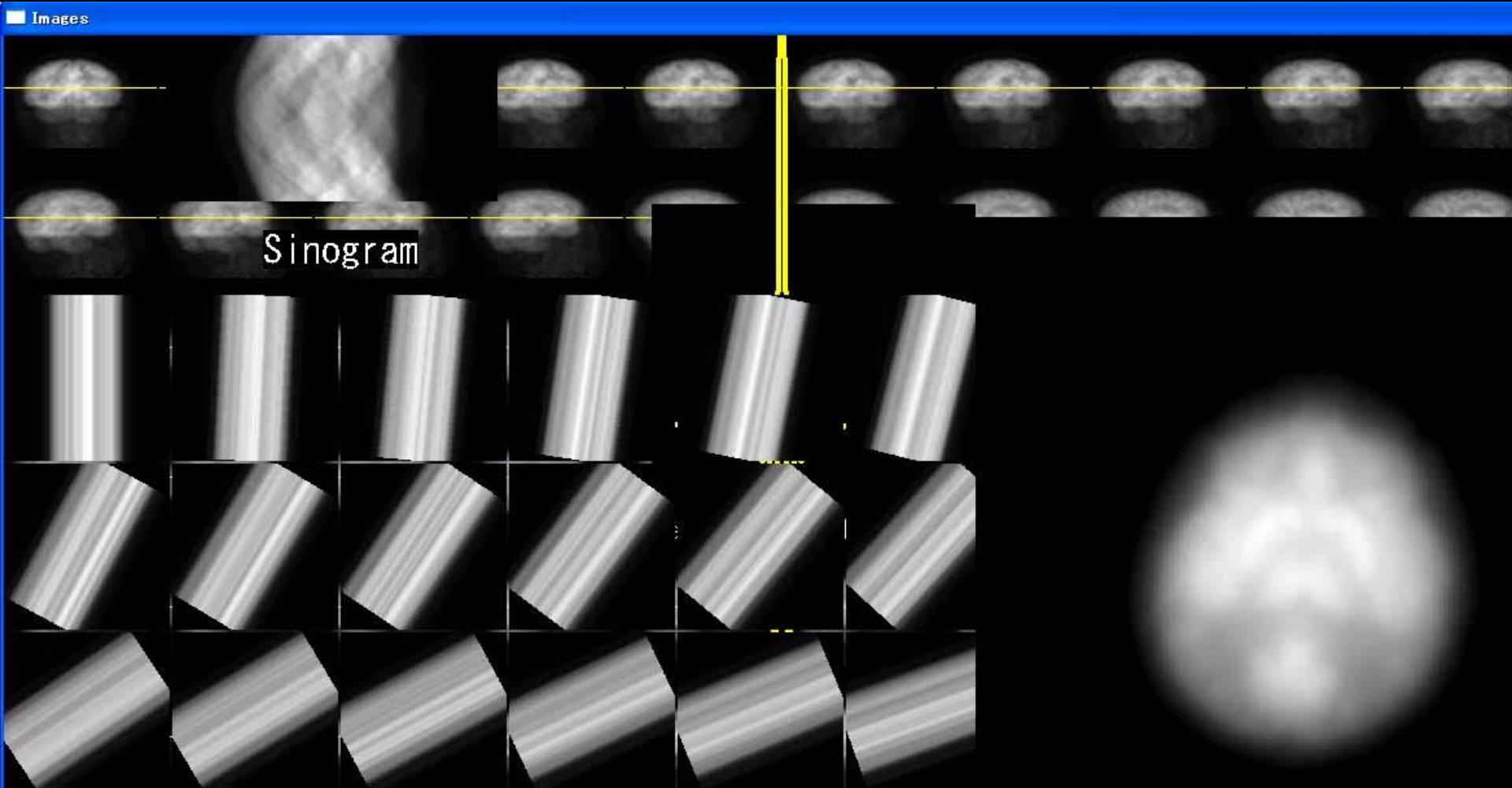
Images



各スライスのサイノグラムが並んでいる3次元データを
並べ替えれば、SPECTのプロジェクション像と同じ並び
になる。 **Sinogram[i][θ][j] = Projection [i][j][θ]**



PETFBP.c を実行して、Simple BackProjection と FBP の再構成画像の違いを観察し、Ramp等のフィルタ関数の必要性を理解してください。

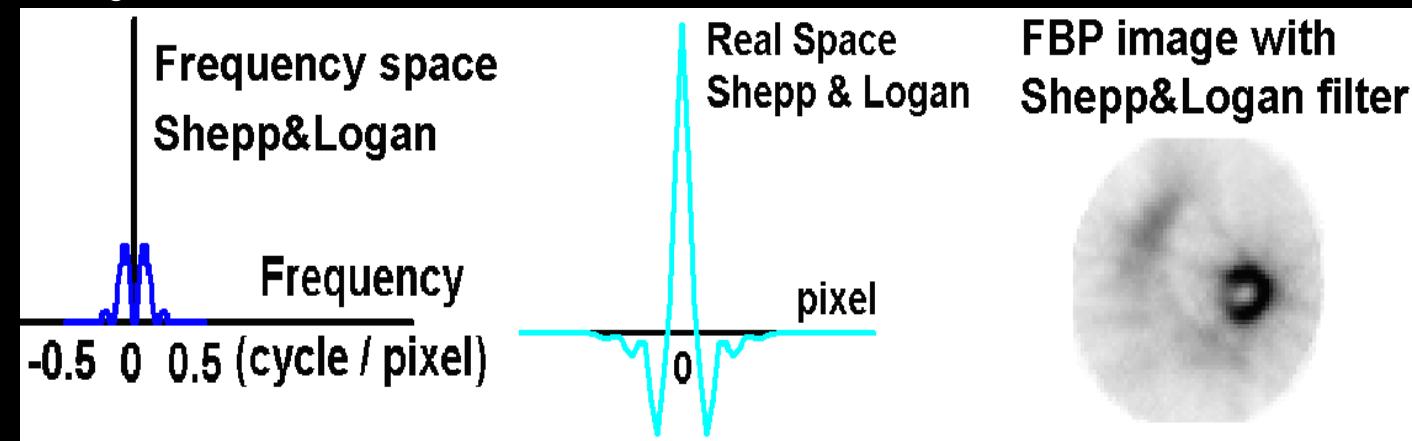


Rampフィルタは理論的には正確な再構成フィルタだが、
実際の臨床データに適用すると ナイキスト周波数以上の高周波
成分を不連続に遮断するために生じる高周波ノイズや 放射状
アーチファクトが目立つ場合が多いため、
臨床では高周波成分を抑制する工夫を施した再構成フィルタが
用いられている。

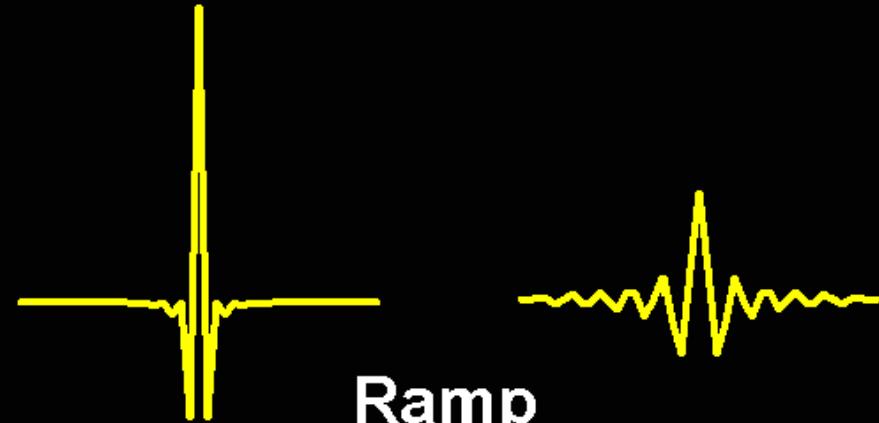
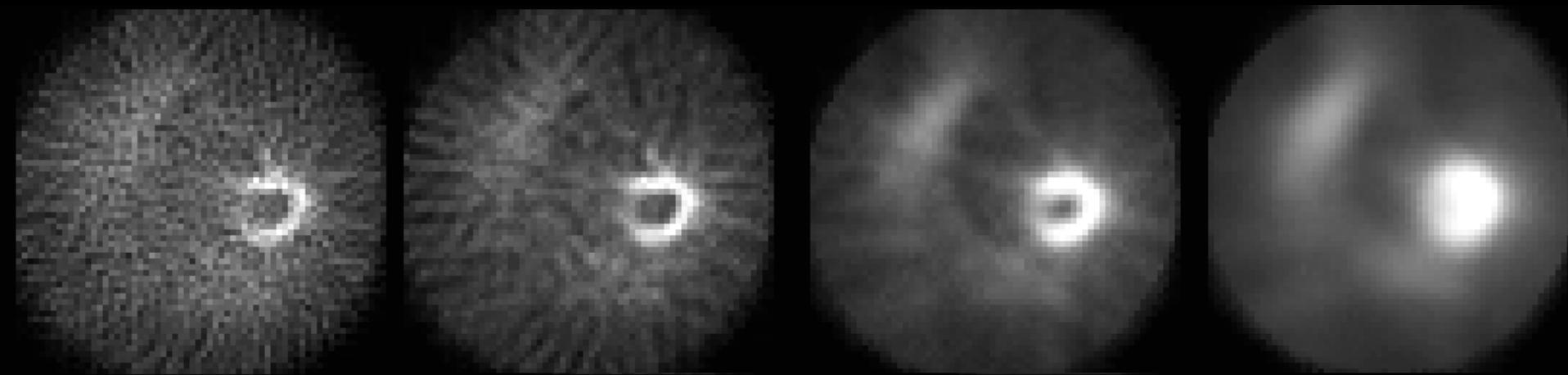
SPECTで用いられる一般的な再構成フィルタとして、
Shepp & Loganフィルタがある。

周波数空間上で、ナイキスト周波数近傍の高周波成分を
連続的に減衰させるように設計されており

再構成画像に生じる**高周波ノイズや放射状アーチファクトを
抑制する効果をもつ。**



実空間フィルタを畳んだ2次元透視画像($\bar{P}\theta$)を重ね合わせると フィルタ逆投影再構成像ができる。フィルタの形状で、再構成画像の高周波成分が変る。



Ramp

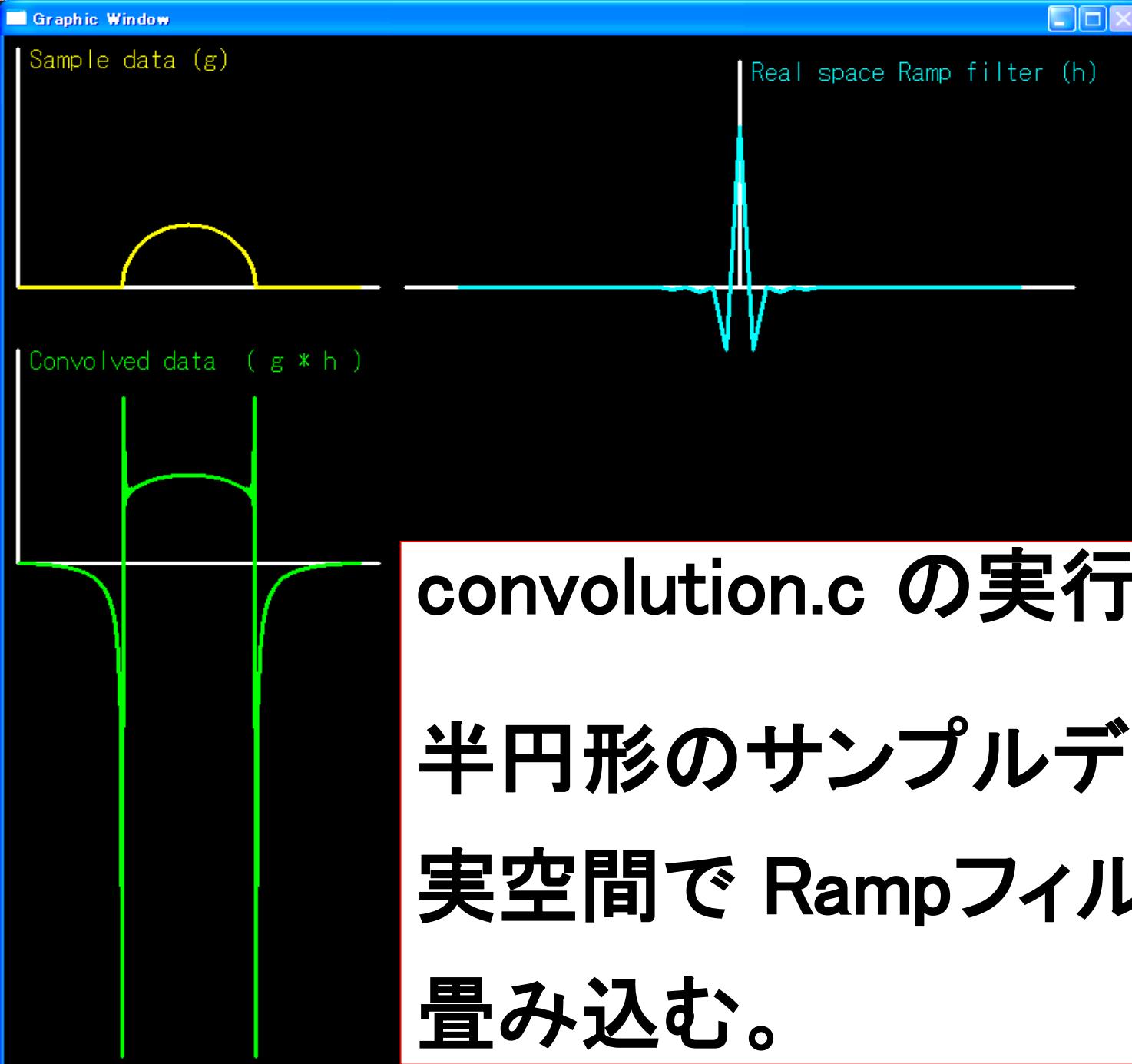
Cutoff = 0.5

Cutoff = 0.3

Shepp&Logan

Cutoff = 0.5

Cutoff = 0.3

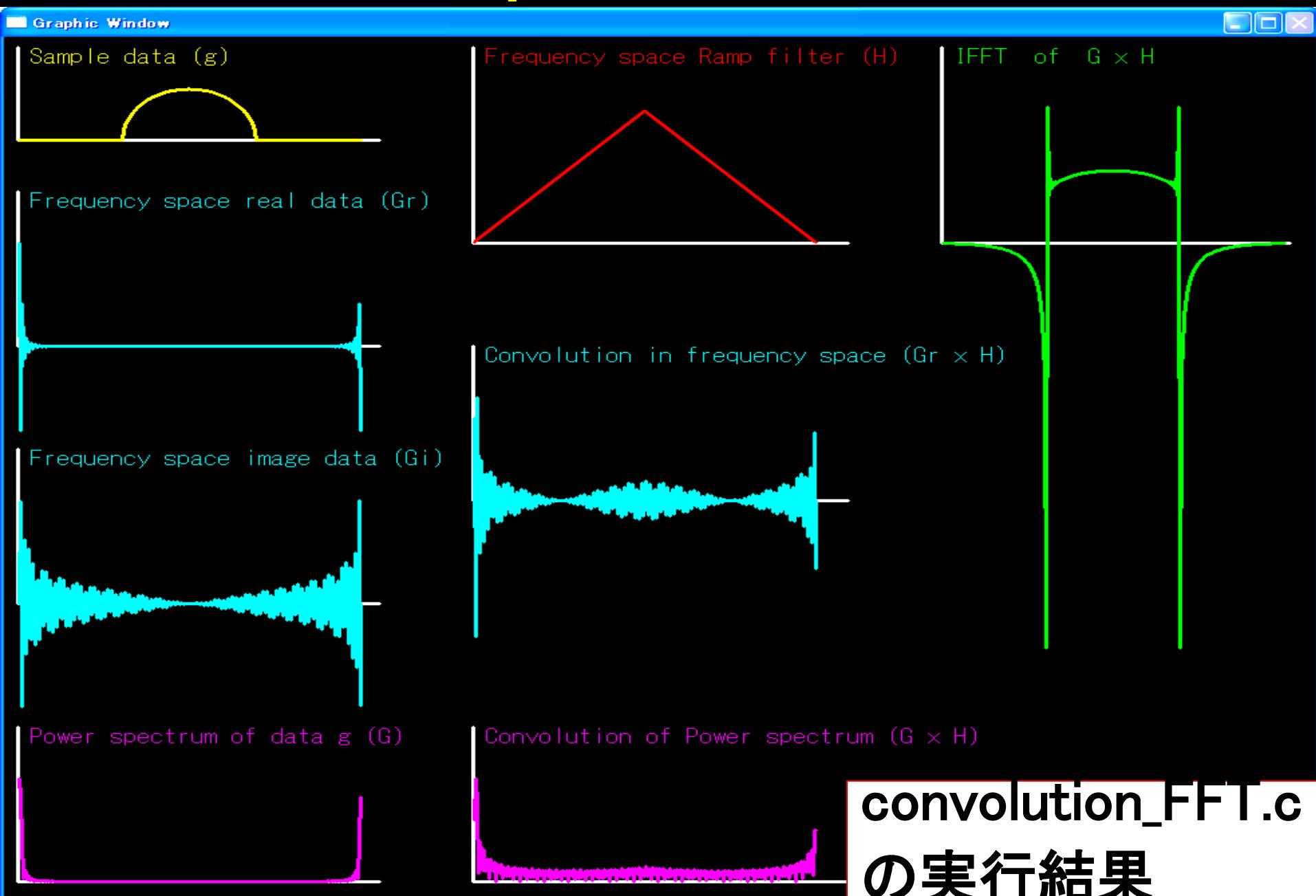


畳み込み Convolution $I = g * h$

サンプルデータ $g[]$ に 実空間Rampフィルタ
 $h[]$ を畳み込んで、配列 $I[]$ に書き込む。

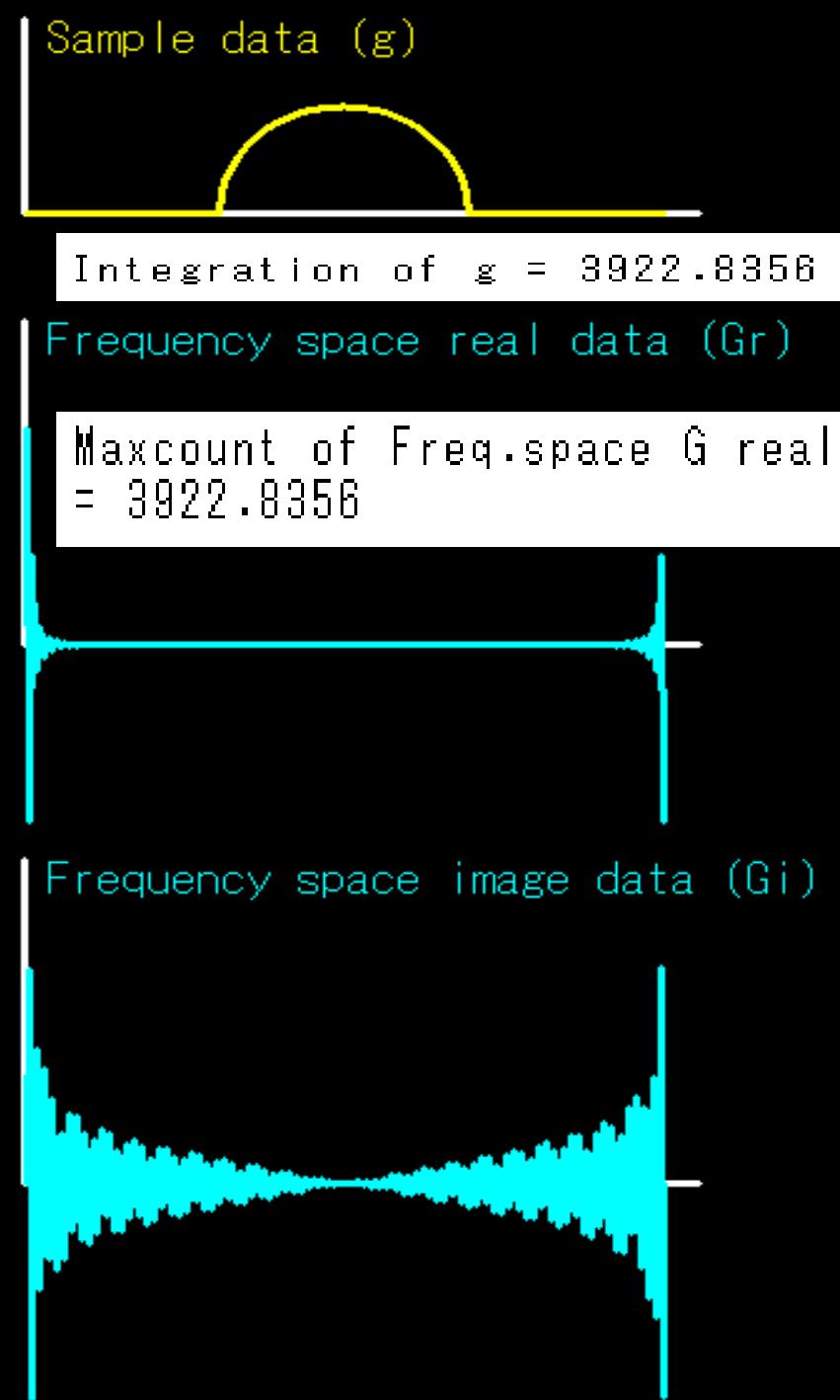
```
//----- Convolution h[ ] into g[ ] -----  
  
printf("¥n¥n Convolution Real space Ramp filter "); scanf("%c",&yn);  
  
for( i=0; i<=255 ; i++ ) {  
  
    gh = 0.0 ;  
  
    for(j=0; j<=127; j++){ if( i+j < 256) gh += g[ i+j ] * h[ j ]; }  
    for(j=1; j<=127; j++){ if( i-j >= 0 ) gh += g[ i-j ] * h[ j ]; }  
    I[i] = gh ;  
}
```

周波数空間で Rampフィルタをかける

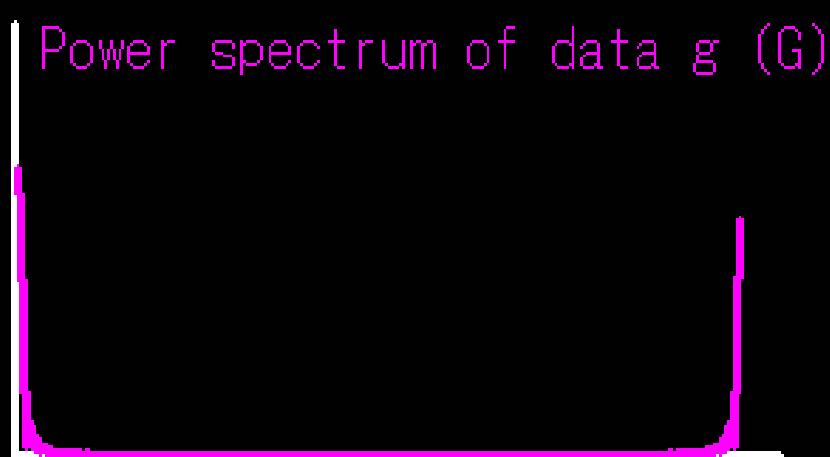


実空間 サンプルデータ g を
フーリエ変換して、
周波数空間の 実数成分 Gr 、
虚数成分 Gi を表示。

実空間 サンプルデータ g の
積分値 が、周波数空間の
実数成分 Gr の最初の成分
(原点、直流成分)と同じ値に
なることを確認する。



周波数空間の 実数成分 G_r 、
虚数成分 G_i の二乗和の
平方根(パワースペクトル G)
を表示。



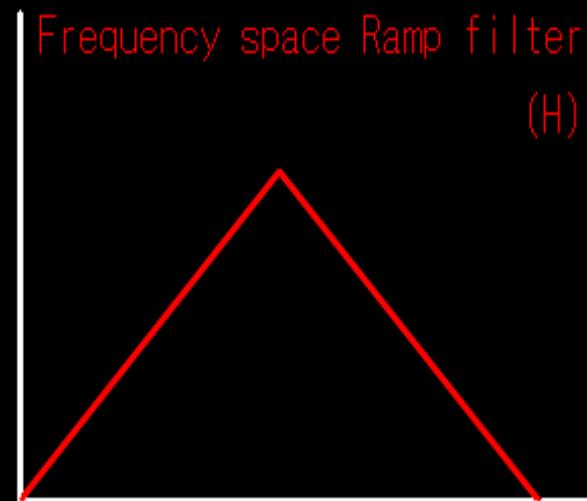
Maxcount of Power spectrum of G
= 3922.8356

実空間 サンプルデータ g の
積分値 が、周波数空間の
パワースペクトル G の最初の
成分(原点、直流成分)と同じ
値になることを確認する。

//----- Convolution H[] into G[] -----

```
for (i= 1; i<= 256; i++ ){ GHr[i] = Gr[i] * H[i] ; }
```

```
for (i= 1; i<= 256; i++ ){ GHl[i] = Gi[i] * H[i] ; }
```



//- Transform convolved GH into real space -

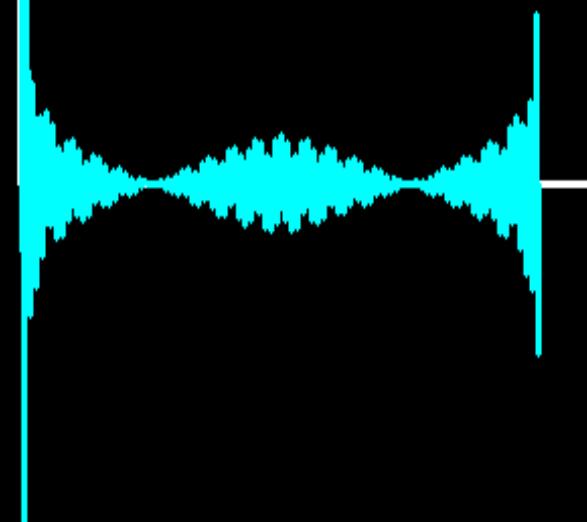
```
FFT1D( GHr, GHl, 256, -1 );
```

Convolution in frequency space
(Gr x H)

データ g の周波数成分 Gr, Gi に、

周波数空間 Ramp フィルタ H をかけて、

逆フーリエ変換する。



データ g の周波数成分 G_r, G_i に、

周波数空間 Ramp フィルタ H を

かけて、逆フーリエ変換すると、

実空間で、実空間 Ramp フィルタ h

を g に畳み込みしたデータと同じに

なる。（ $G \times H$ と $g * h$ は等価演算）

積分値が 元データ g と同じことを

確認する。

（カウント総和は保たれる。）

Integration of Convolved g with FFT
= 3922.8356

IFFT of $G \times H$

