核医学機器工学概論　２

断層画像CT（ Computed Tomography）を得る方法

1. フィルタ重畳逆投影法

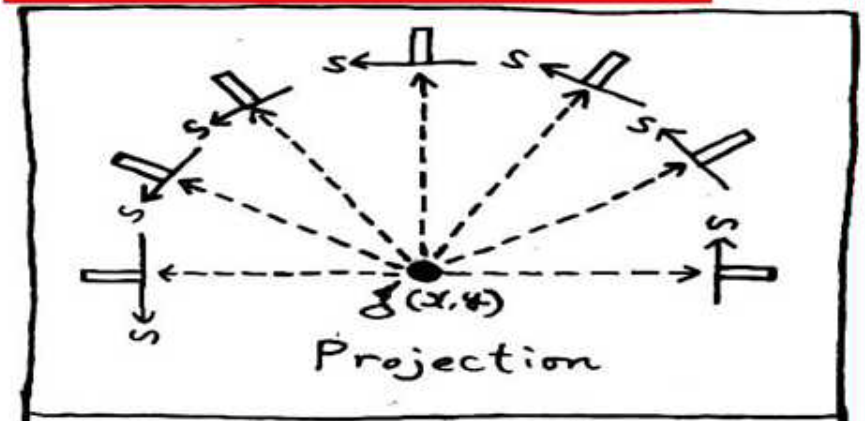
FBP（ Filtered Back Projection ）

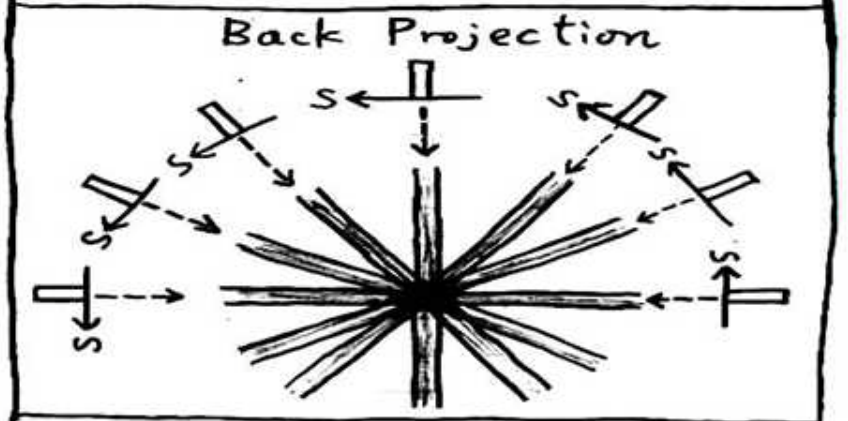
2. 逐次近似再構成法　Iterative Reconstruction

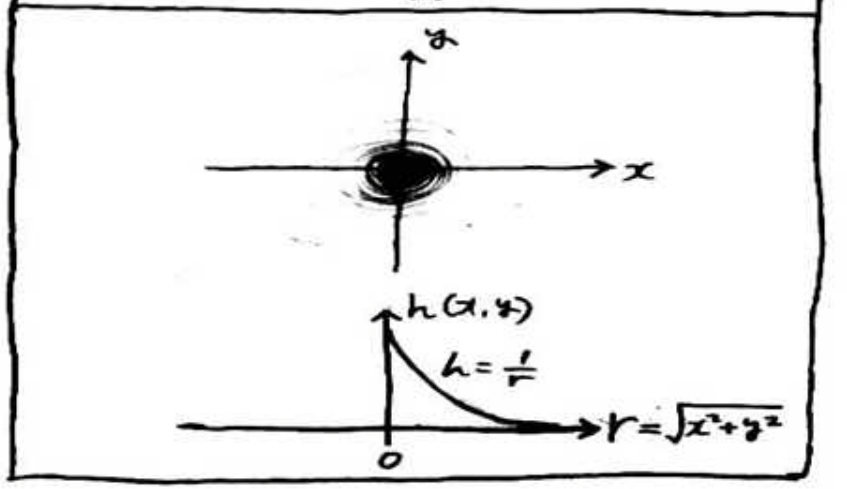
MLEM（Maximum Likelihood Expectation Maximization）

OSEM（ Ordered Subsets Expectation Maximization ）

⑤ Back Projection







xy平面上の1点 $f(x,y)$ の透視データ $P_\theta(s)$ (Projection) を $\theta=0$〜π の範囲で求める.

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って, 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

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

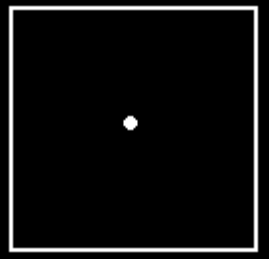
プログラム　PSF.exe の実行。フォルダ PSF 内 の PSF.exe をダブルクリック。

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。

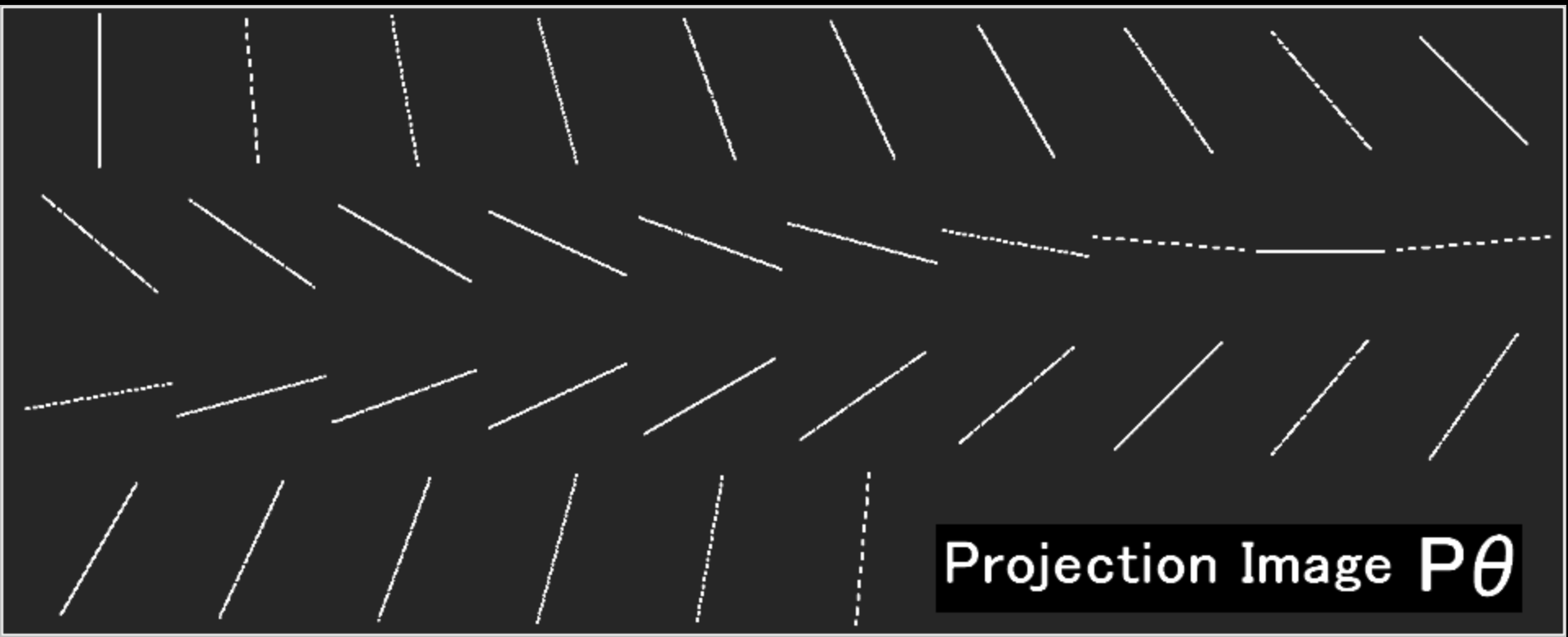
選択する再構成フィルタは、
（real space filter は）
フォルダ PSF 内 にある
RealRAMP256.txt を選択。

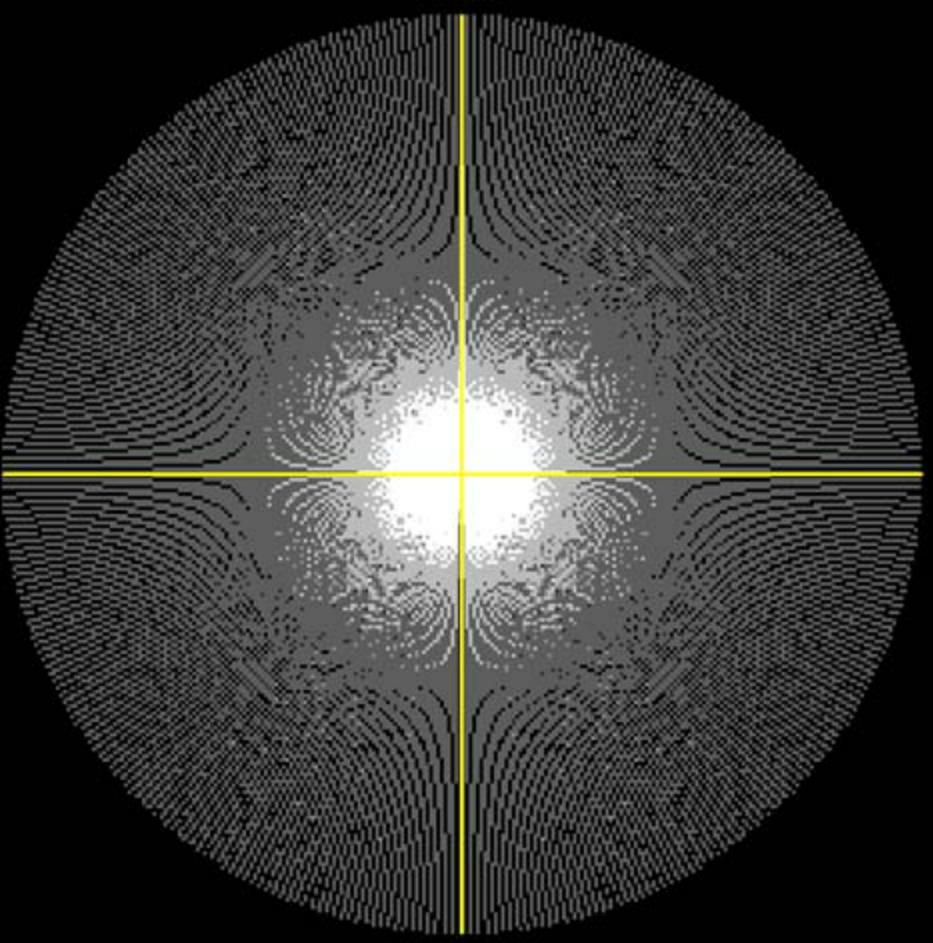
画像中心の１画素だけに画素値を与え、
他を0に設定した2次元画像を作成。



その像を180度方向から横から透視したと想定した
像 Ｐθ 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）

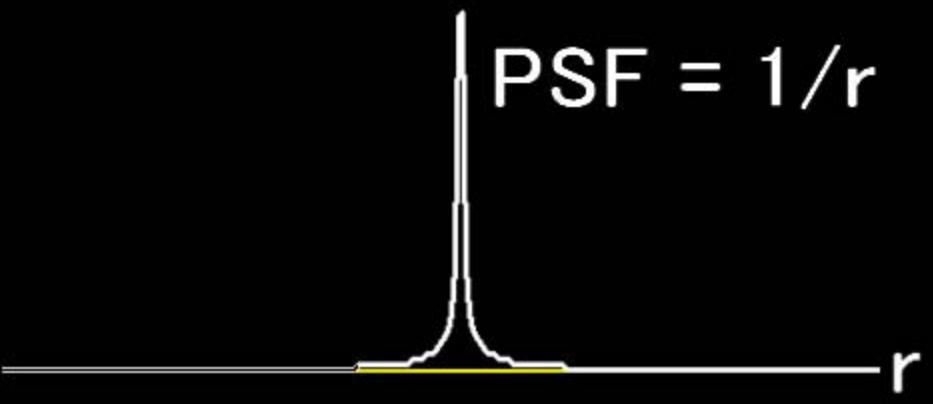




180枚の線状画像 Pθ
を重ね合わせると、
広がりをもつ分布が
得られる。

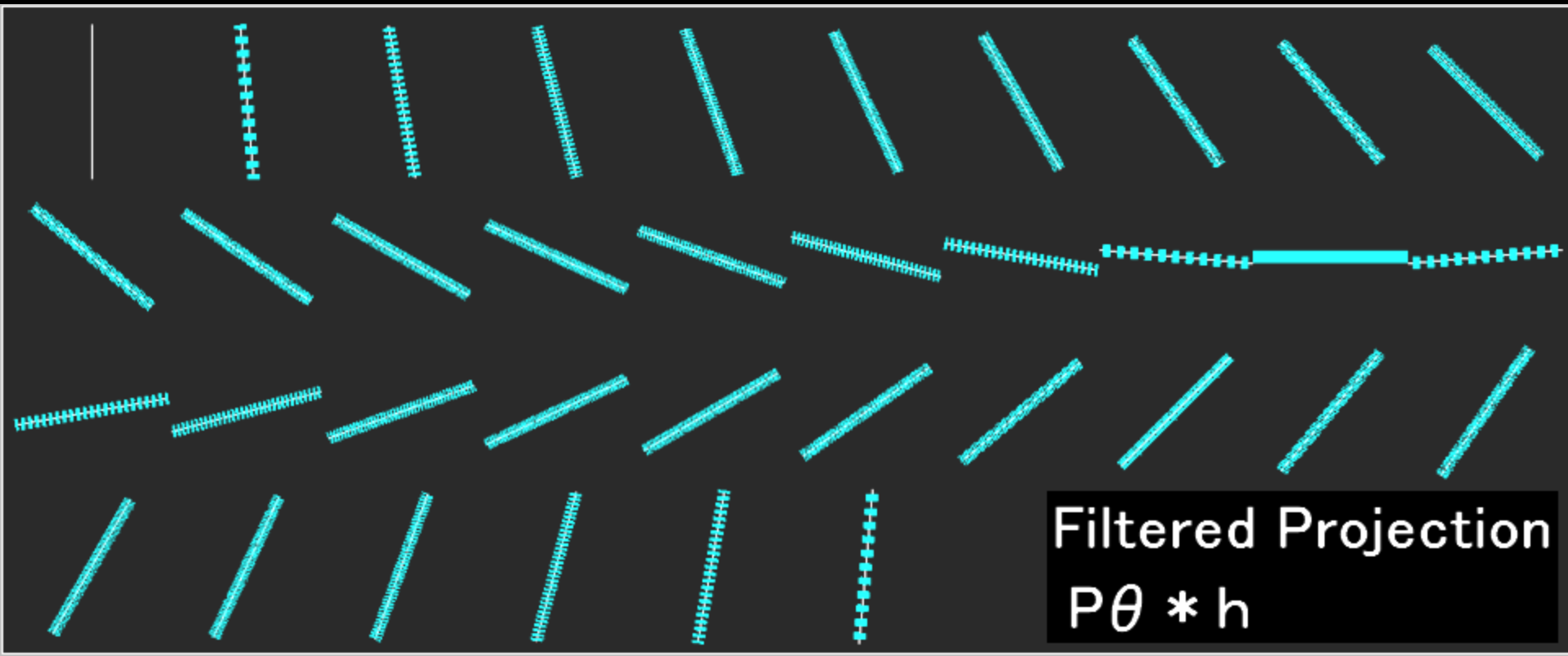
点広がり関数　PSF
(Point Spread Function)

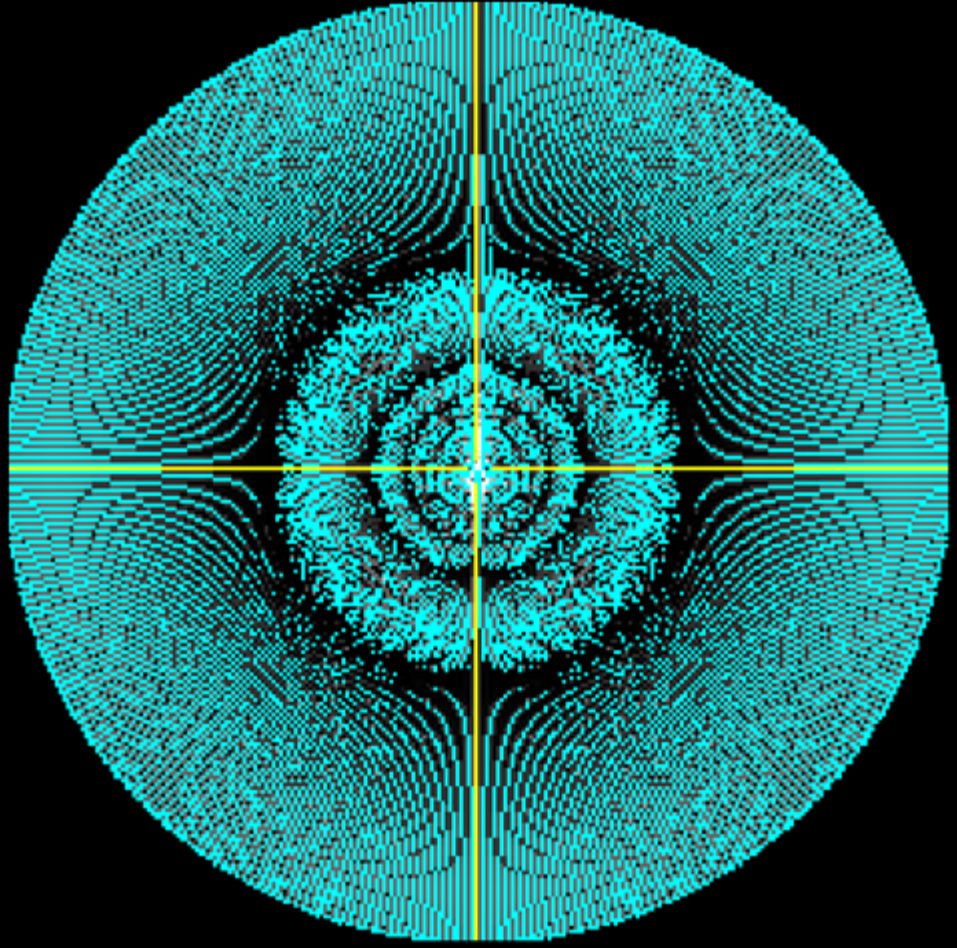
PSF　= 1/r

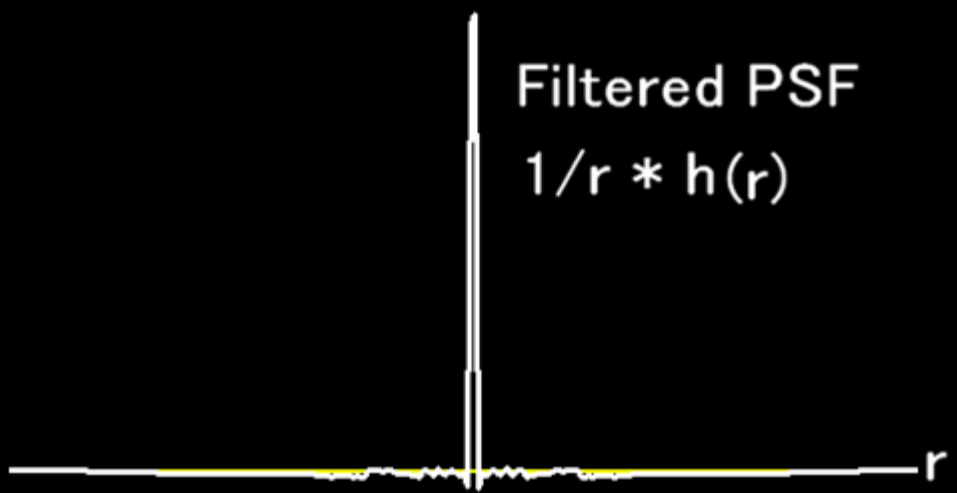


180枚の線状画像 Pθ に、
実空間 Ramp (RL) filter　h を畳み込む（ Pθ ＊h ）。

（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
Pθ＊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

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・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、

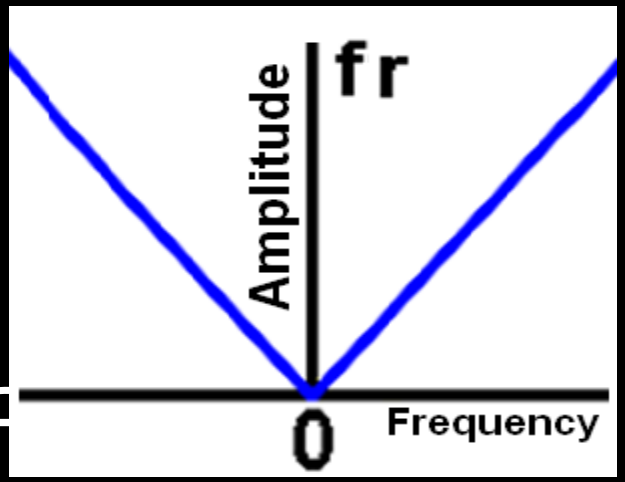
（ｆｒは周波数空間上の原点からの距離　）

Ｆ（１／ｒ）＝

∫∫（１／ｒ）exp（－ｊ（2πｒｆｒ））ｒｄｒｄθ　＝１／ｆｒ

これより　Ｌ　＝　Ｇ　／ｆｒ　なので　Ｇ＝Ｌ・ｆｒ

Ｇ ＝Ｌ・ｆｒ　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 ｆｒ（ ｆｒ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・ｆｒ　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　　（ ｈ は フィルタｆｒ の 逆フーリエ変換 ）

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

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

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

$$g = \int \overline{P\theta} \, d\theta \quad \left(\overline{P\theta} = P\theta * h \right)$$ **FBPの式**

Pθ 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 ＝｜ｆｒ｜（ｆｒがナイキスト周波数未満の場合）

H ＝　０　　（ｆｒ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。

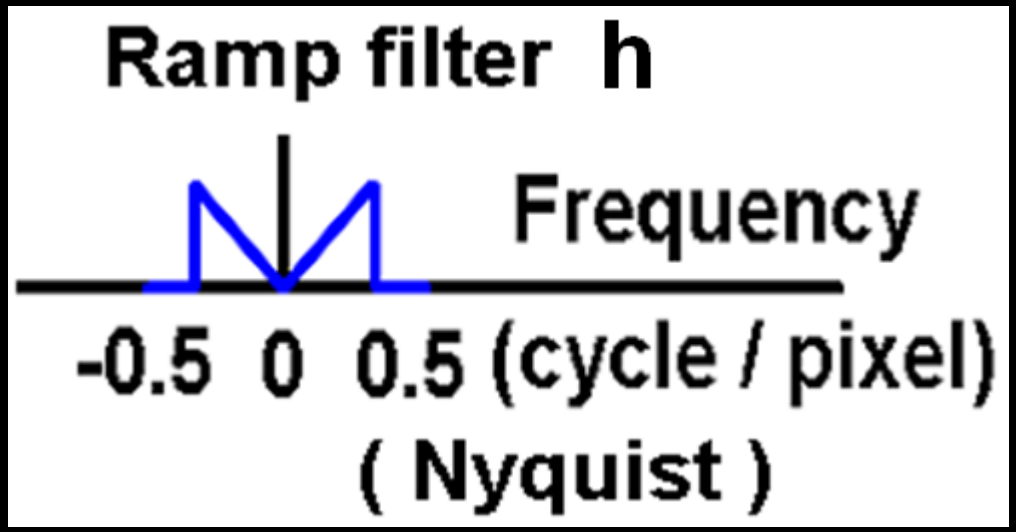
Ｒａｍｐフィルタを逆フーリエ変換して

実空間Ｒａｍｐフィルタｈにしてから、

実空間でPθにｈを畳み込む。

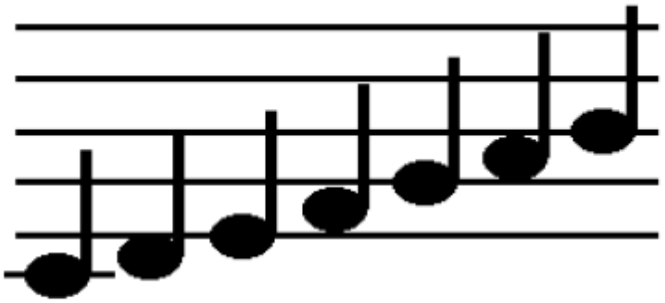
$$\overline{P\theta} = P\theta * h$$

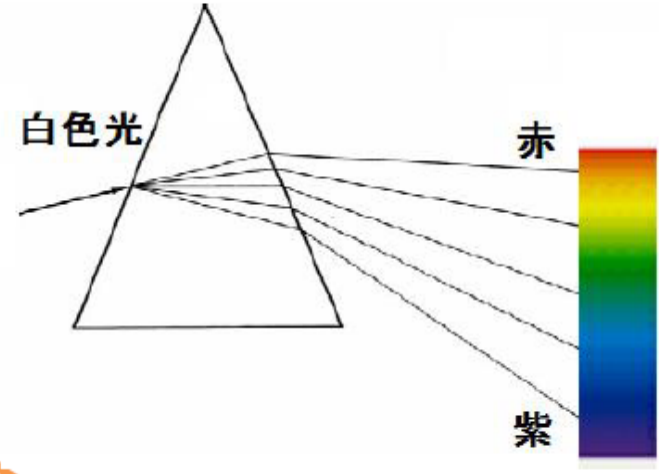
（＊は畳み込み演算）



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 $\int\int g(r) \exp(-j \cdot 2\pi r\, fr)\, r\, dr\, d\theta$ （ヤコビアンに注意）
$g(r) = 1/r$ とすると $\int\int (1/r) \exp(-j \cdot 2\pi r\, fr)\, r\, dr\, d\theta = \int\int \exp(-j \cdot 2\pi r\, fr)\, dr\, d\theta = 1/fr$





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。

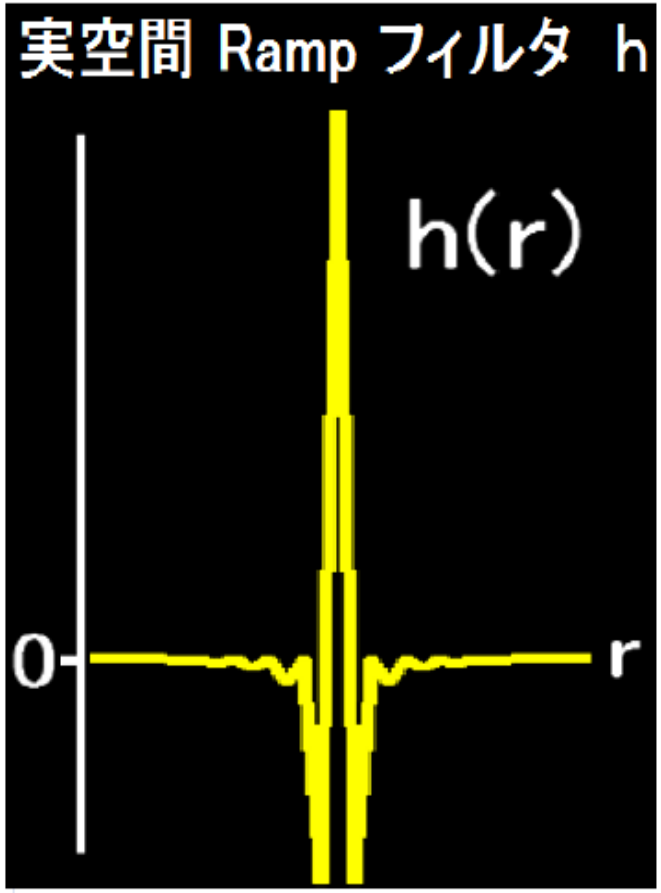
虹は白色光に含まれる色の周波数分布を示している。雨粒中で屈折しやすい高周波の紫色成分から、屈折しにくい低周波の赤色成分まで、周波数の順序で色が並んでいる。

基礎

Rampフィルタ

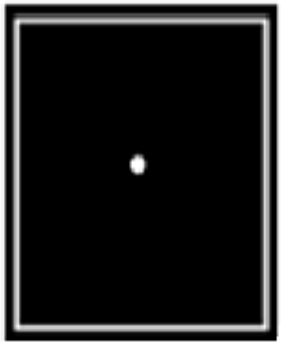
周波数空間でのフィルタ frとは、どのような形をしているのか。１次元周波数空間の縦軸の周波数成分が fr の値を持つフィルタであるが、横軸が周波数軸すなわち fr である。つまり周波数空間で、正比例のような形状になる。

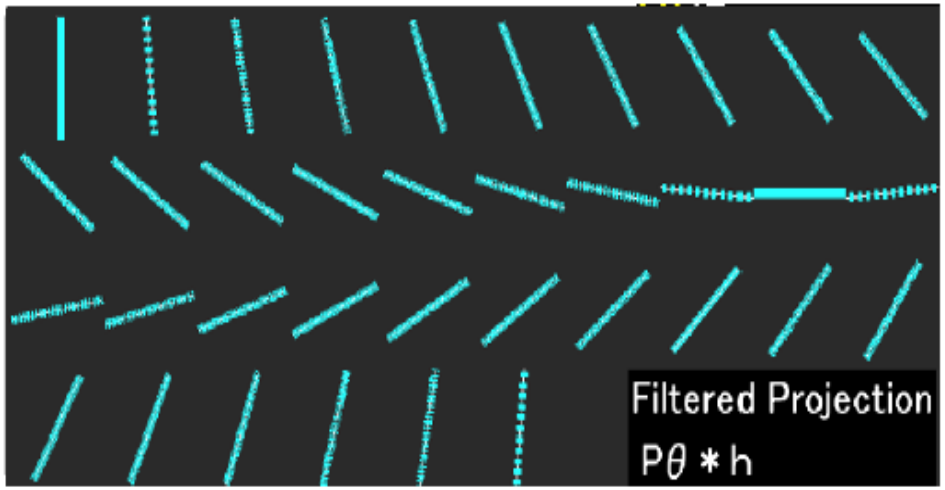
周波数にはマイナスの値は無いが、高速フーリエ変換（FFT）のアルゴリズムの都合で見かけ上、マイナスの周波数データが存在する。縦軸の周波数成分にはマイナスは無いので、周波数がマイナス側の成分は絶対値となり、原点を中心に左右対称な絶対値 |fr| の関数として表現される。
さらにサンプリング定理によって、ナイキスト周波数（0.5 cycle / pixel）以上の高周波成分はノイズなので除去する。
よって、このフィルタは上図のようになり、これをRampフィルタという。

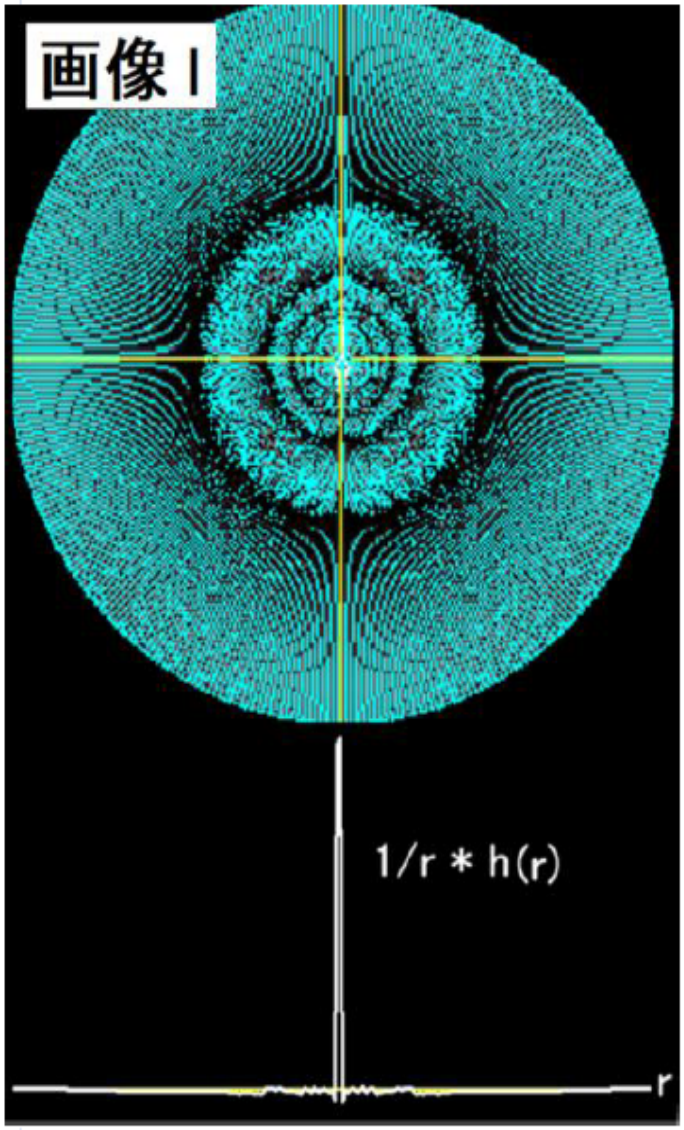


周波数空間の Rampフィルタを H と表現する。
H を 1 次元逆フーリエ変換すると、実空間での
Rampフィルタ h が左図のように算出される。

画像 g







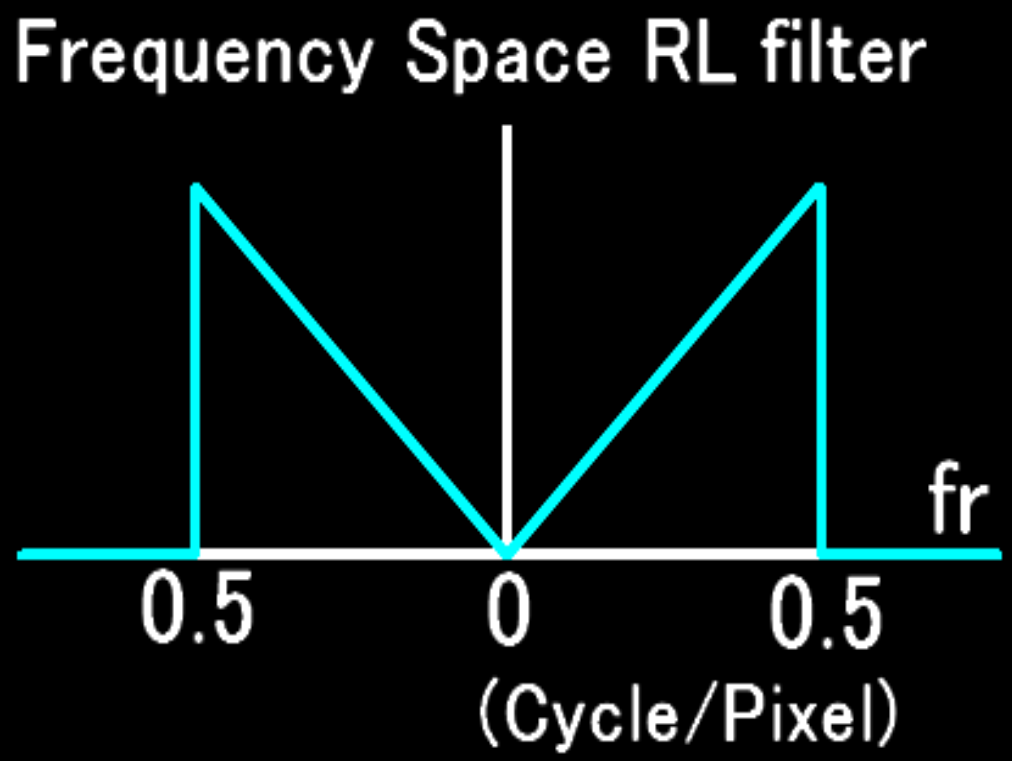
画像中心の１画素だけ値が１で、他は全て0 の256x256画素の画像 g を1度毎に180度方向から透過投影した投影像Pθを作成する。次に全ての Pθ に Rampフィルタhを畳込む（式で表すと Pθ ＊h）（上図で青色の部位はマイナスの画素値）（全てのPθ ＊hの断面が hと同じ形状をしている）。さらに、全ての Pθ ＊h を重ね合わせて画像 I を算出する。式で表すと $I = \int (P\theta * h)\, d\theta$

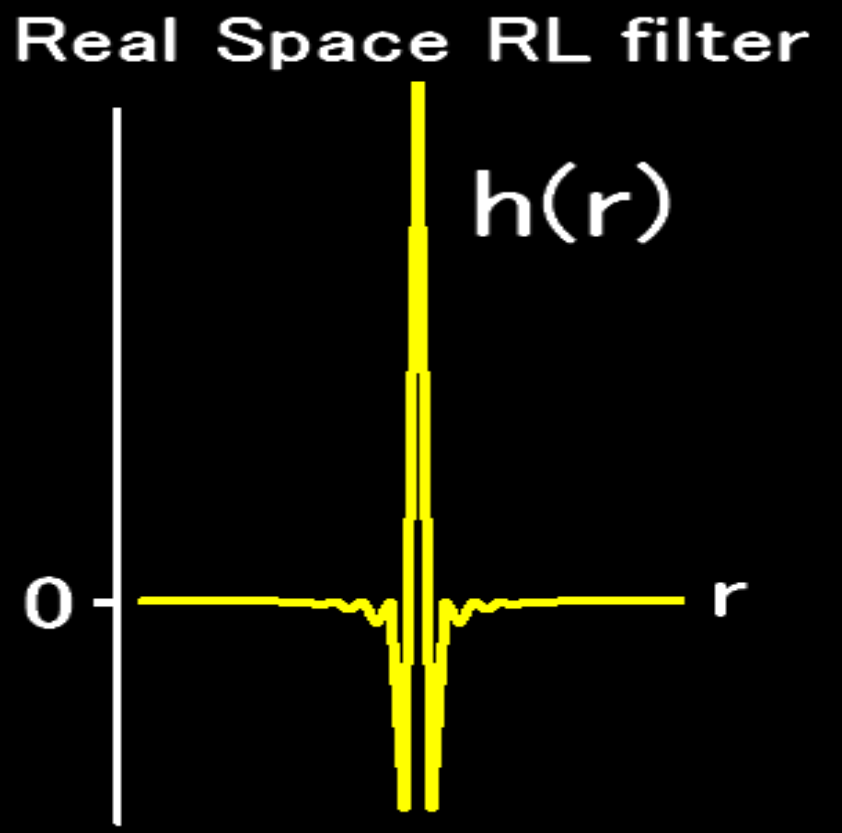
画像 I は中心の１画素だけ値が１で、周囲は殆ど0 に近い値を示し、ほぼ画像 g に戻っている。

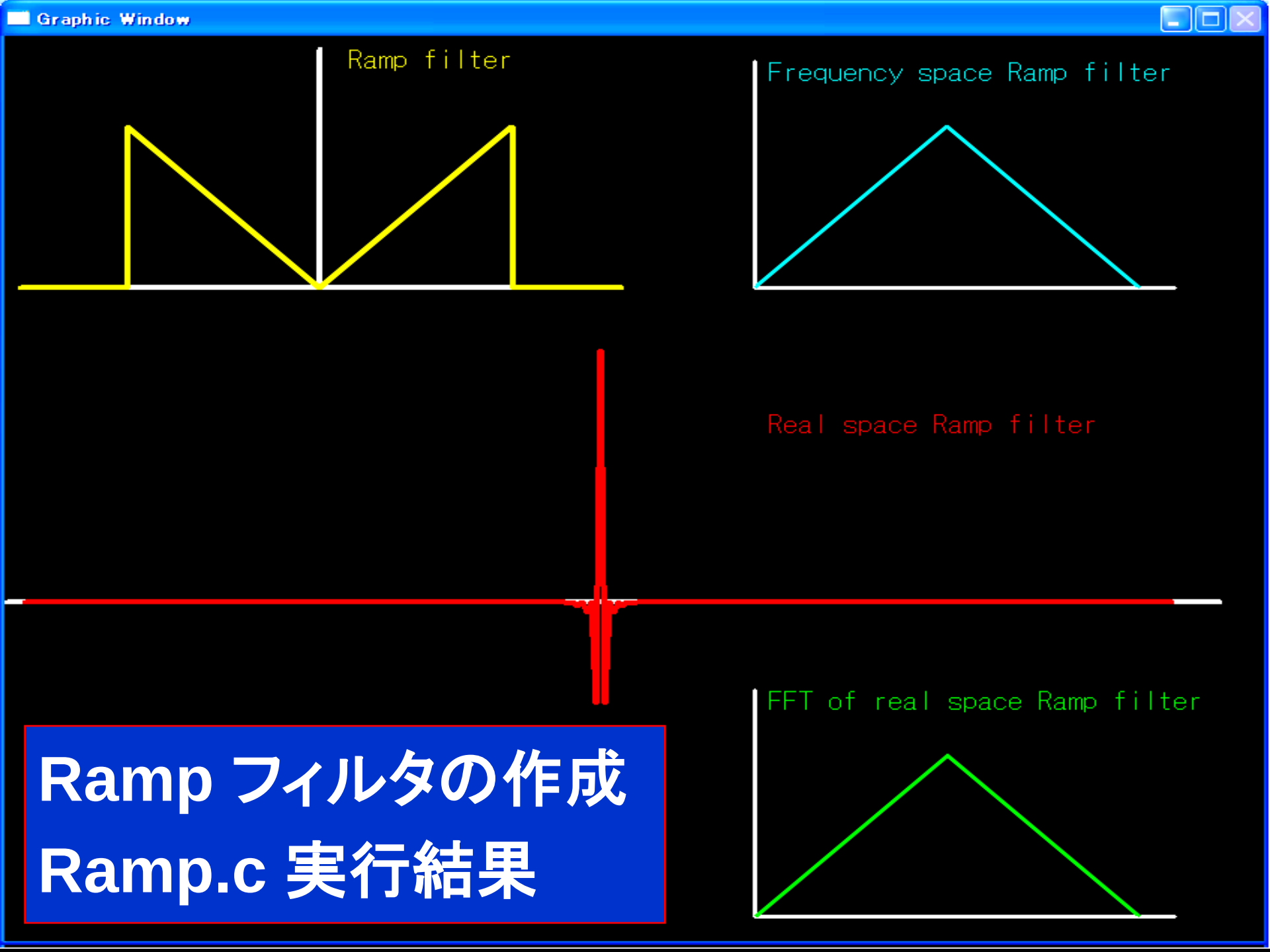
Pθの単純重ね合わせ像では、画像中心の１画素が最大値を示し、その近傍に中心からの距離 r に反比例する1/r の関数に画素値が広がっていたが、Rampフィルタhを畳込むと、その広がりが補正されて元の画像が再現される。このRampフィルタhの機能によって断層画像を算出することができる。

周波数空間 RL フィルタ（ 1／fr ）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。

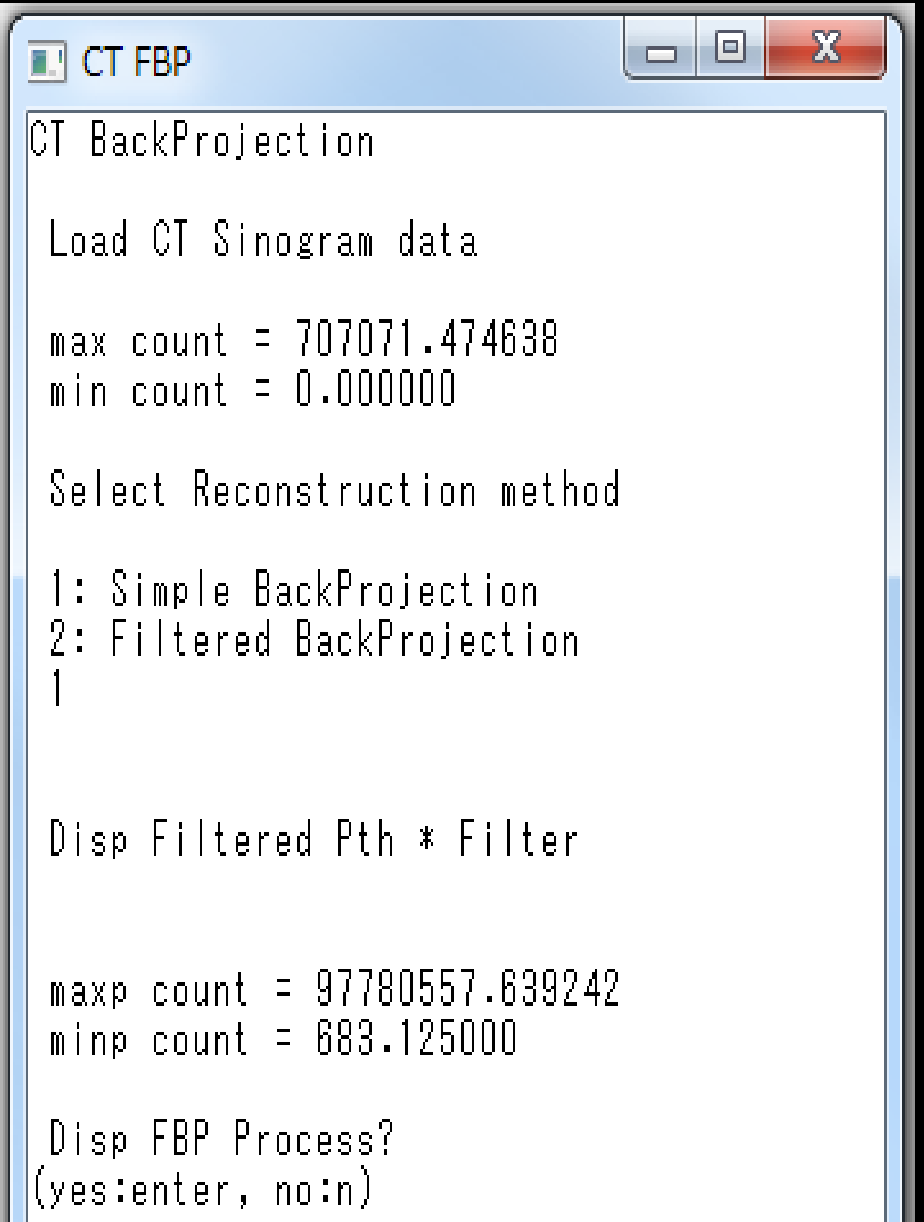






Ramp フィルタの作成
Ramp.c 実行結果

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。

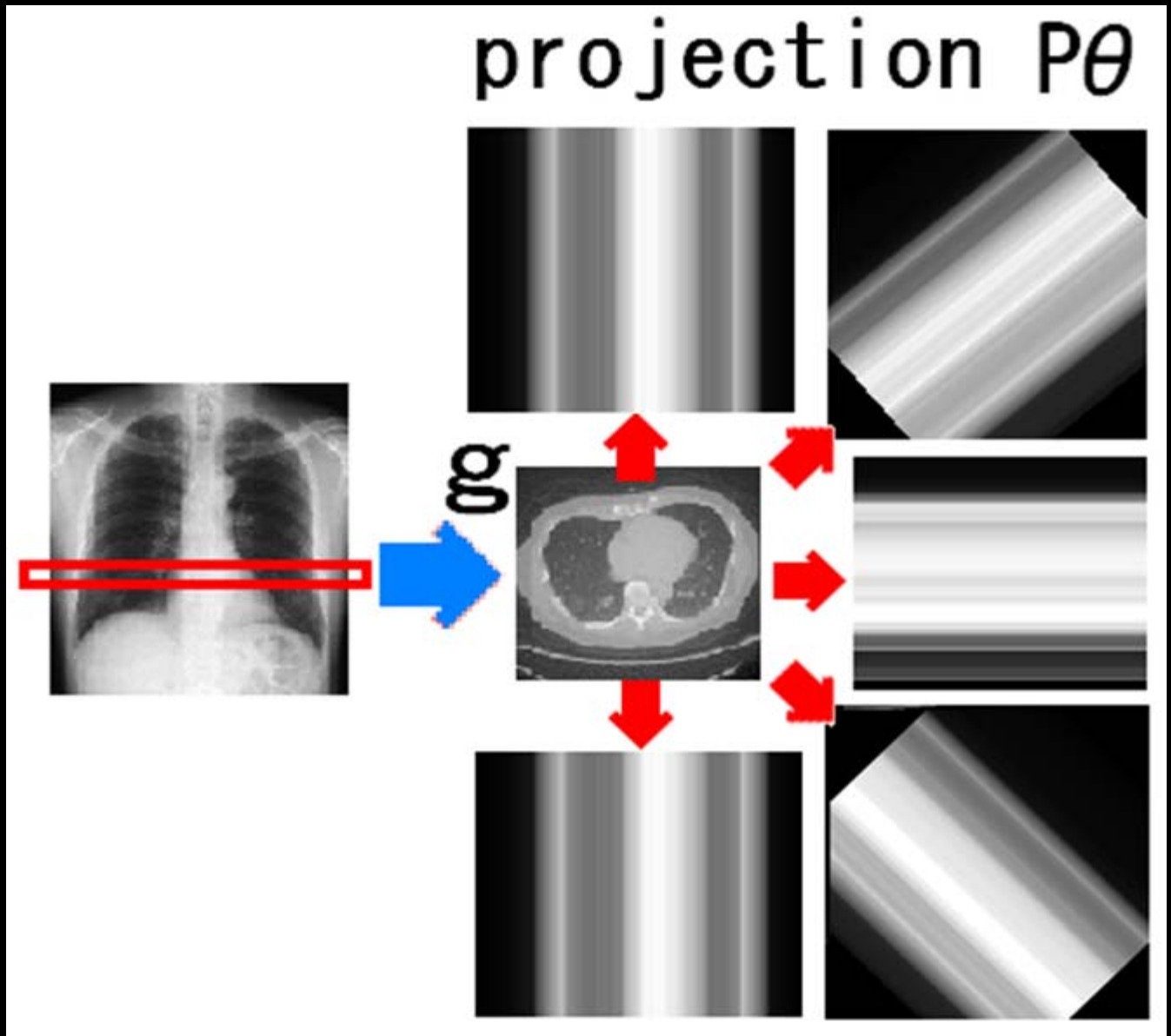


このテキストウィンドウ内をクリックする。
選択するプロジェクションデータは、フォルダ CT 内 の CTprojection を選択。

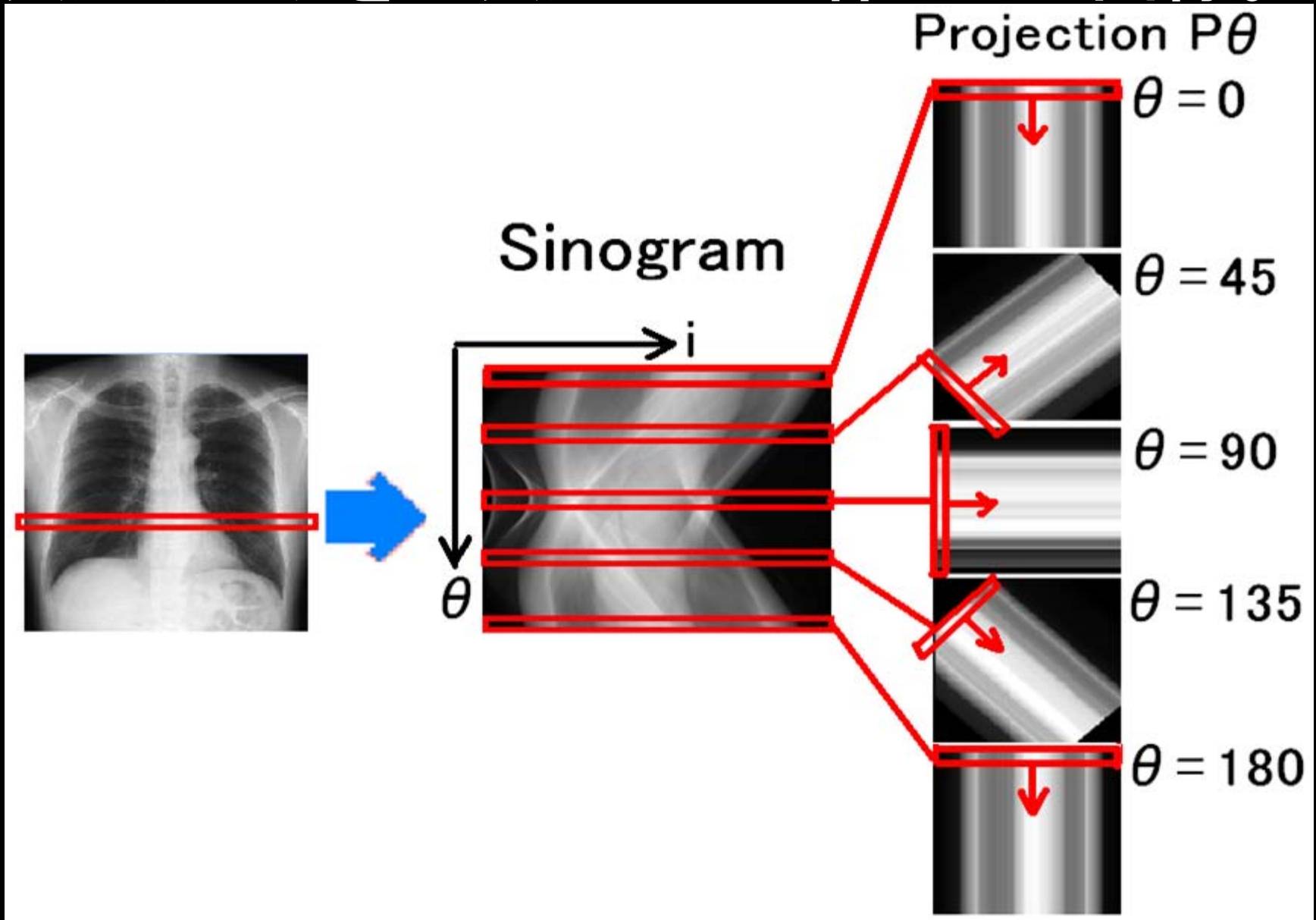
1を入力して
Simple back projectioを実行。

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 Pθ を測定。

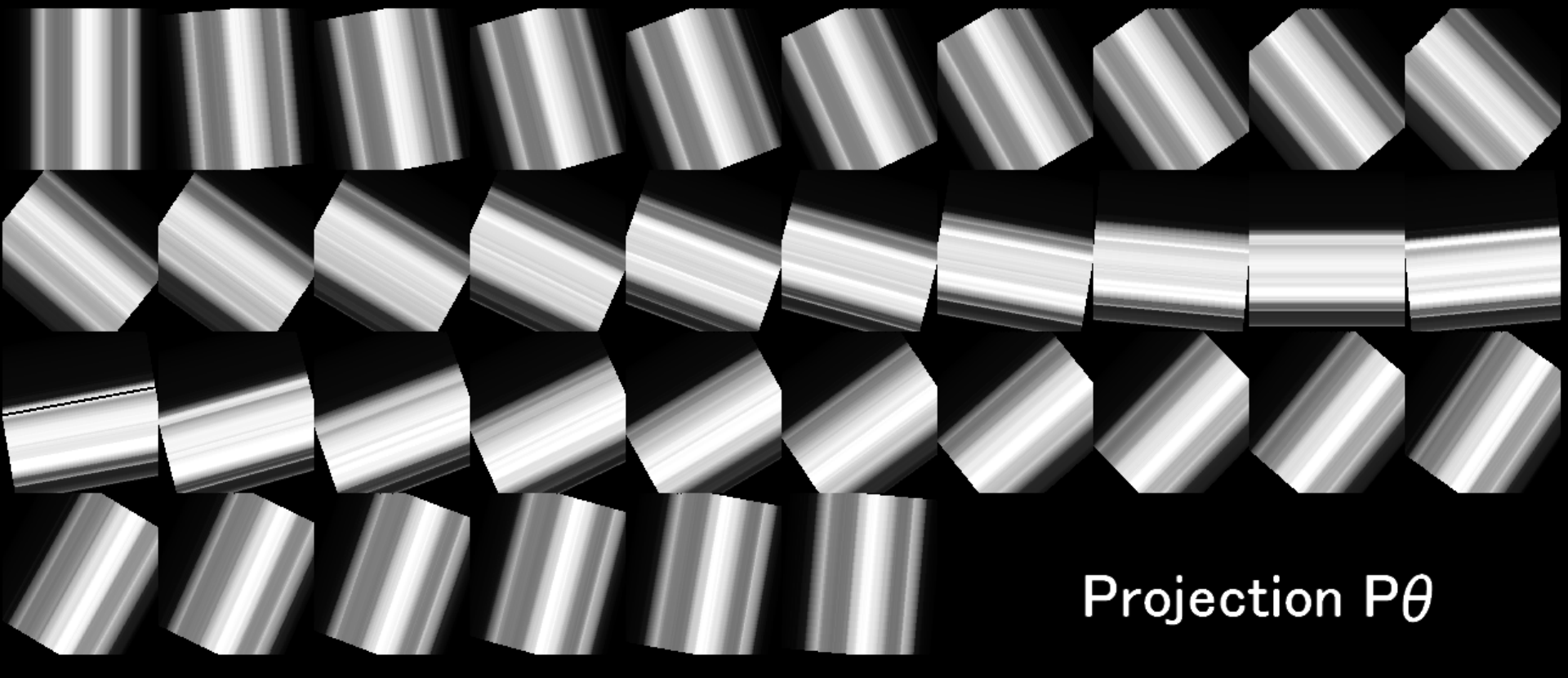


2次元透視像 Pθ は、サイノグラムの各行 θ の1次元データを 2次元に引き伸ばした画像。



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 Ｐθを
作成（ １度ごと、合計１８０枚の２次元透視画像 ）。

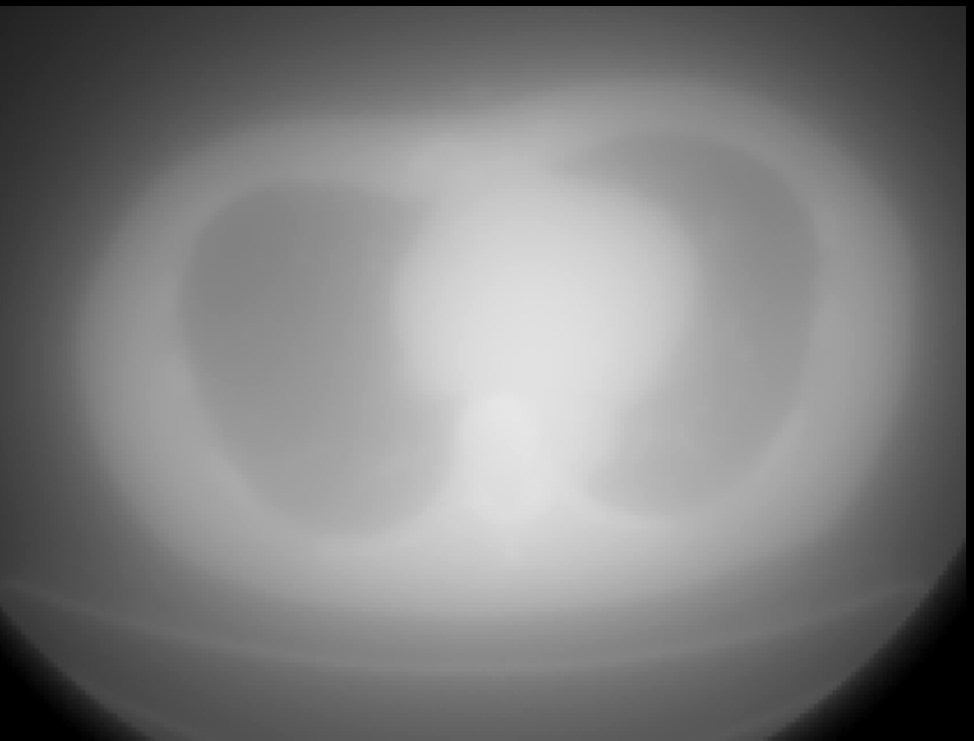
（ スライドでは５度ごとの画像を表示）



180度方向から透視した像 Ｐθ 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 畳み込まれ
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(Pθ)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。)

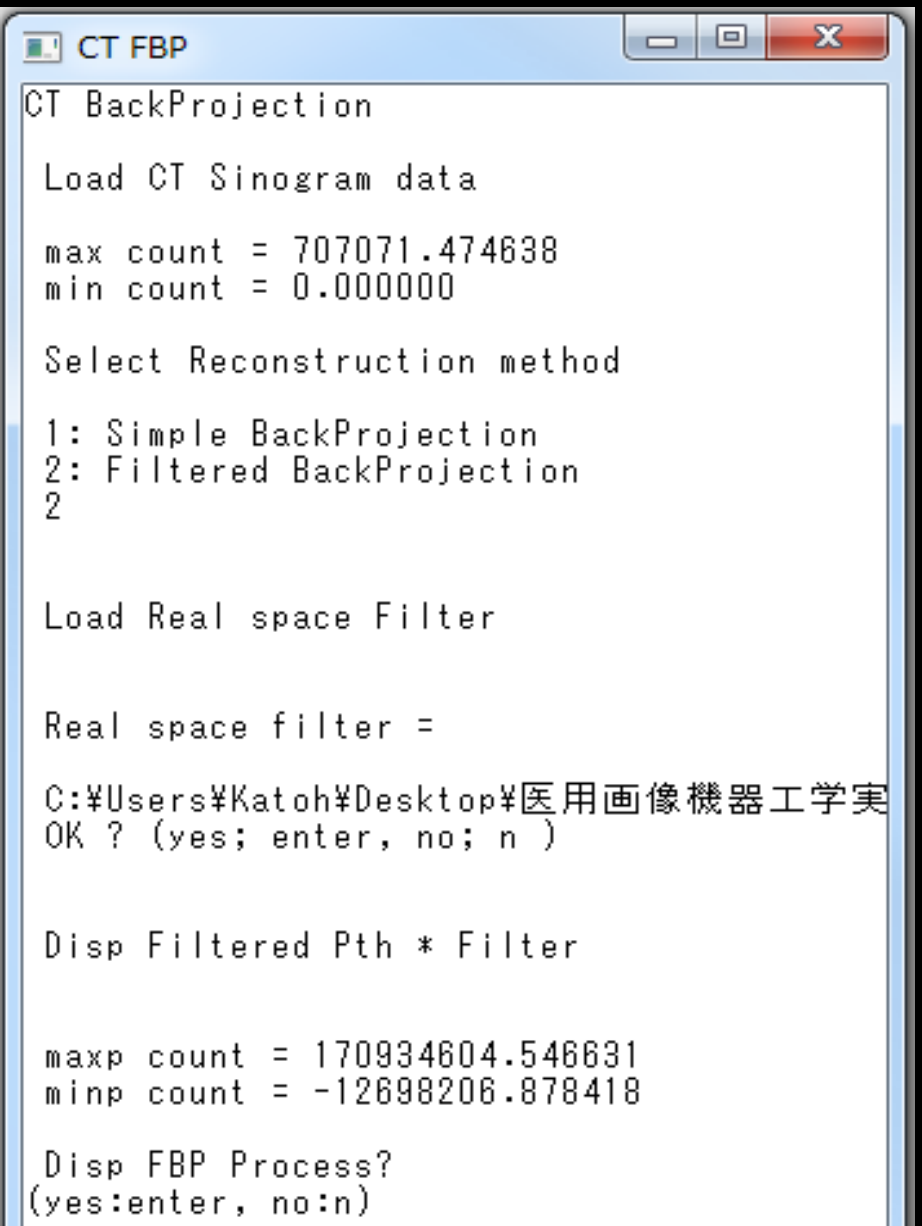
スライス j におけるサイノグラムを求める。
サイノグラムの各スライスの１次元配列は、
各々の角度から収集されたデータ。

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Pθを作成する。
Pθ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。



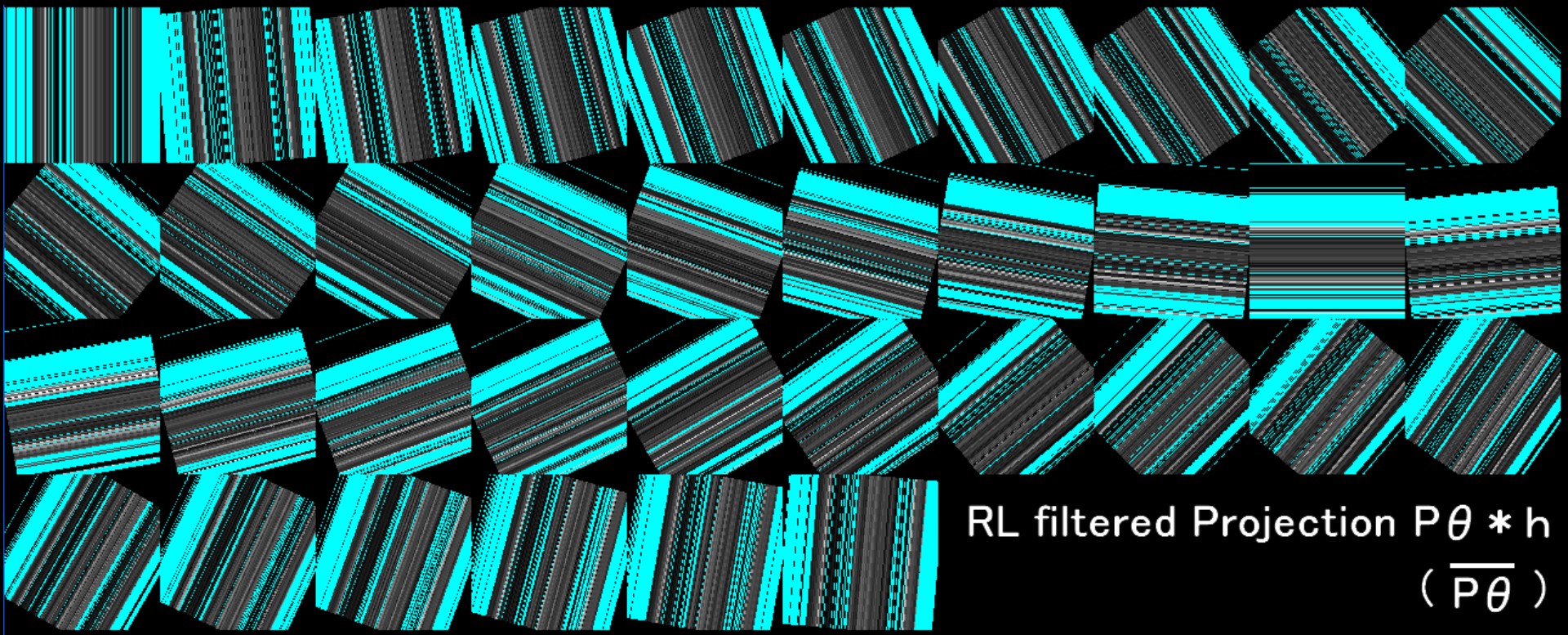
このテキストウィンドウ内をクリック。
選択するプロジェクションデータは、フォルダ CT 内 の CTprojection を選択。

2を入力して
Filtered back projectio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
($P\theta * h$)。

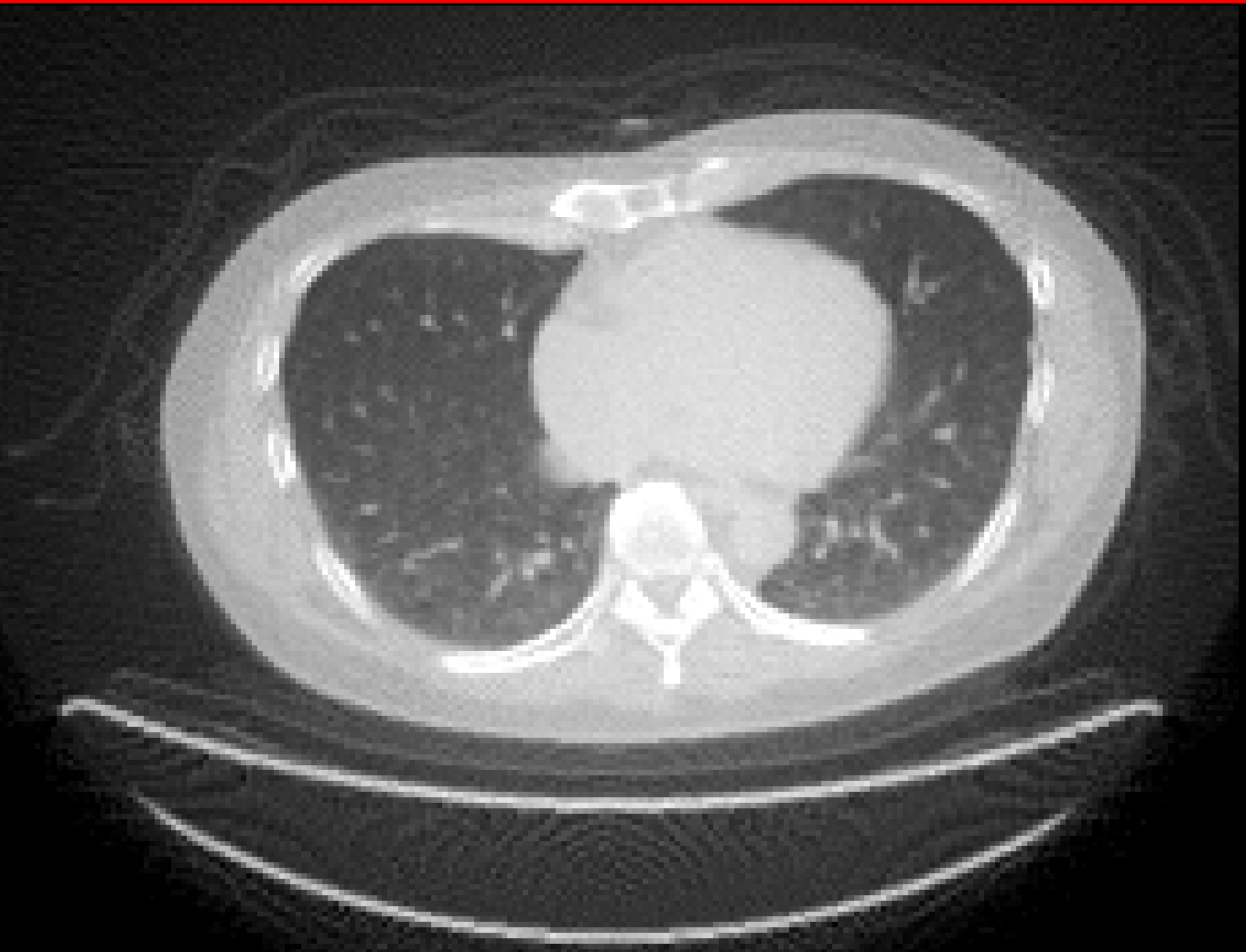
(青く表示された画素はマイナス値)



$P\theta * h$ を重ね合せると 正確な断層像 g になる。

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

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 x 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

$$l(x) = \int_{-\infty}^{\infty} f(n)\, h(x-n)\, dn$$

Convolution 積分
（たたみ込み積分）

これを $l(x) = f(x) * h(x)$ と表わす。

$h(x)$ を convolution 関数という。

$f(x)$ が $h(x)$ のために $l(x)$ に ボケてしまったことを意味する。

たたみ込みの定理

$h(x-n)$ の Fourier 変換を $H(f)$ とすると

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

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

$$l(x) = \int f(n) \left[\int H(f)\, e^{j(2\pi f x)}\, e^{-j(2\pi f n)}\, df \right] dn$$

$$= \int \left[\int f(n)\, e^{-j(2\pi f n)}\, dn \right] H(f)\, e^{j(2\pi f x)}\, df$$

$$= \int \boxed{G(f)}\, H(f)\, e^{j(2\pi f x)}\, df$$

$l(x)$ の Fourier 変換を $L(f)$ とすると

$l(x) = \int L(f)\, e^{j(2\pi f x)}\, df$, よって $\boxed{L(f) = G(f)\, H(f)}$

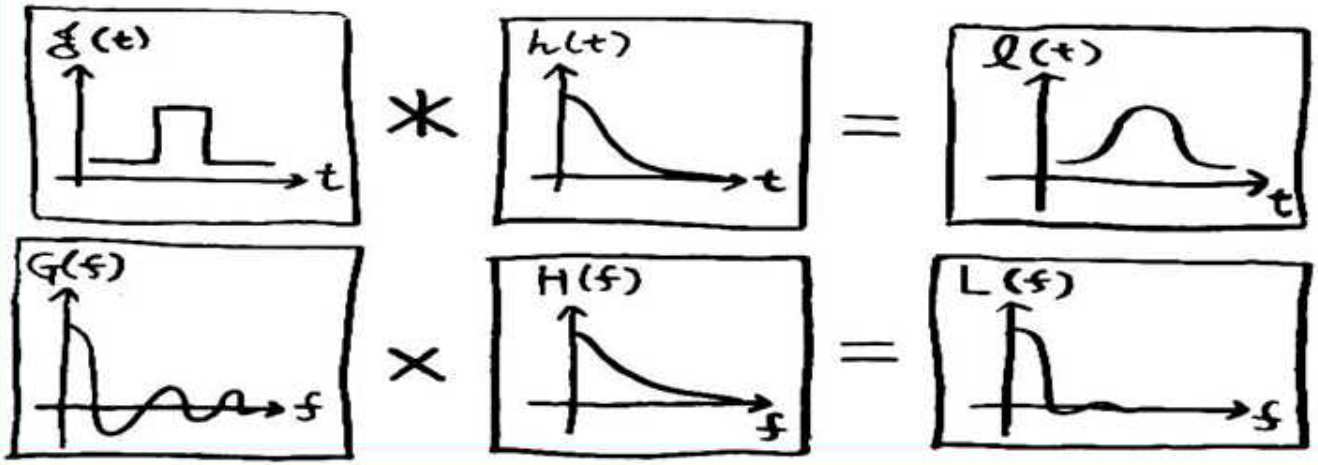
Date 20

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，

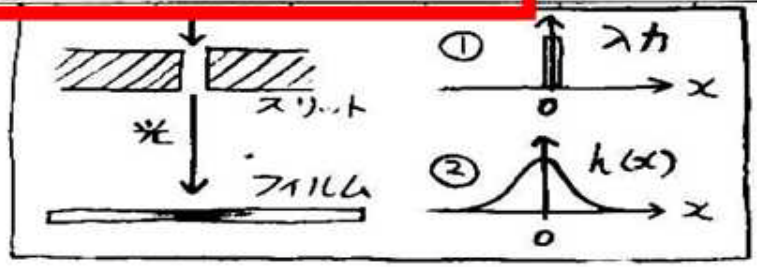
曲線gの細かい変動（高周波成分）がhによって
減衰して大ざっぱな曲線lになることを示す．

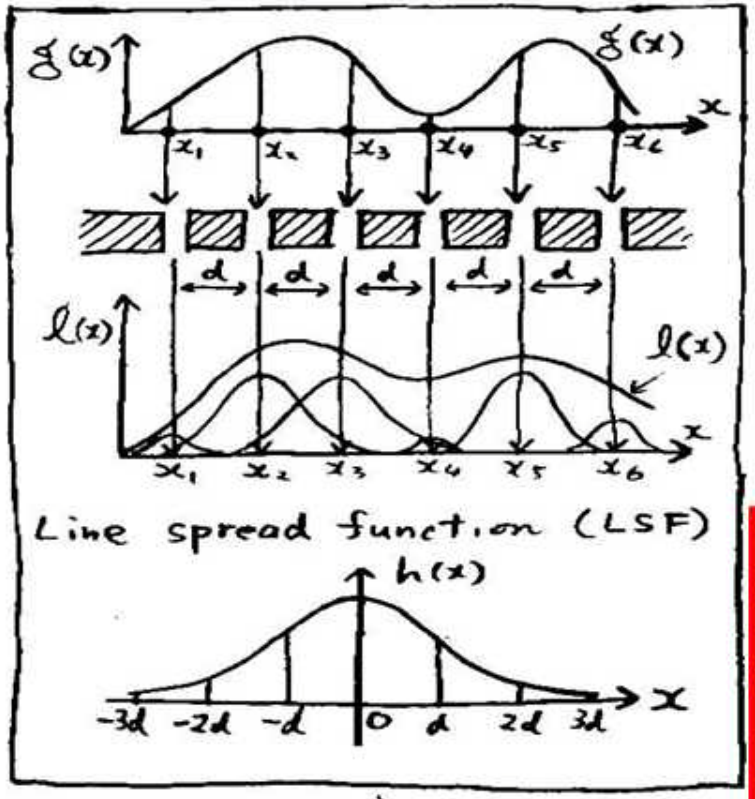
つまり周波数空間において，G(f)はH(f)によって
高い周波数の成分だけ減衰させられて
高周波成分の少なくなったL(f)になる．

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って，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

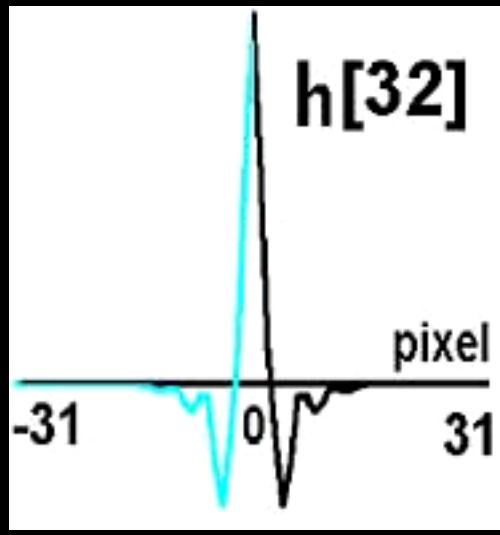
$$l(x_4) = g(x_4)h(0) + g(x_5)h(-d) + g(x_6)h(-2d) + g(x_3)h(d) + g(x_2)h(2d) + g(x_1)h(3d)$$

任意の座標x_iにおける$l(x_i)$は

$$l(x_i) = g(x_i)h(0) + g(x_{i+1})h(x_i - x_{i+1}) + g(x_{i+2})h(x_i - x_{i+2}) + \cdots + g(x_{i-1})h(x_i - x_{i-1}) + g(x_{i-2})h(x_i - x_{i-2}) + \cdots$$

$$= \sum_{n=-\infty}^{\infty} g(x_n)h(x_i - x_n)$$

データ数64の1次元データ g[0]～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]～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 I [0]～ I [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] , I[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 ] ;  }

        I[i] = gh ;
    }
```

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

l[i] = g[i]*h[0] + g[i+1]*h[−1] + g[i+2]*h[−2] + g[i+3]*h[−3] + ・・・
　　+ g[i−1]*h[1] + g[i−2]*h[2] + g[i−3]*h[3] + ・・・

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、

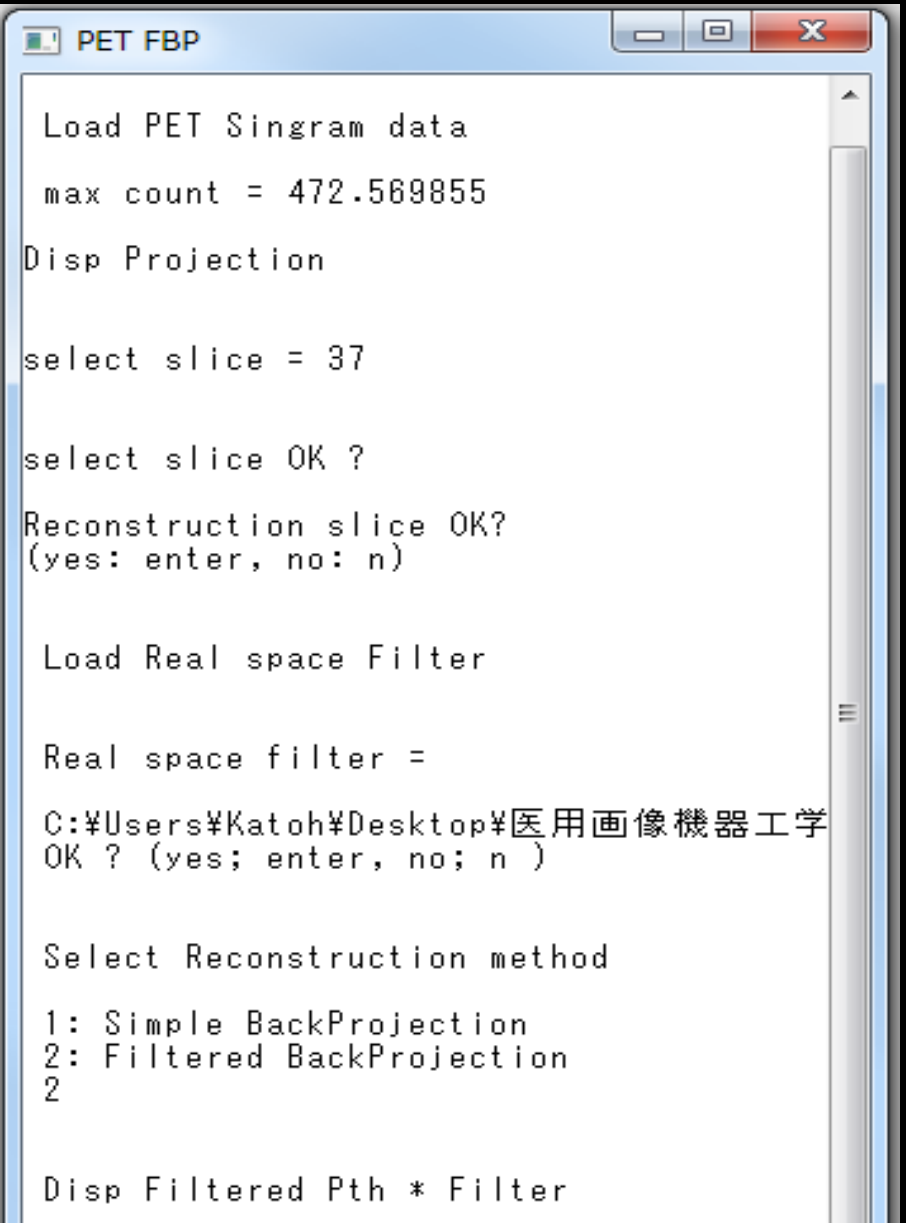
l[i] = g[i]*h[0] + g[i+1]*h[1] + g[i+2]*h[2] + g[i+3]*h[3] + ・・・
　　+ g[i−1]*h[1] + g[i−2]*h[2] + g[i−3]*h[3] + ・・・

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 ] ;  }
l [ i ] = gh ;
```

これを i が 0 から 63 まで繰り返すと、l [0] から l [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。



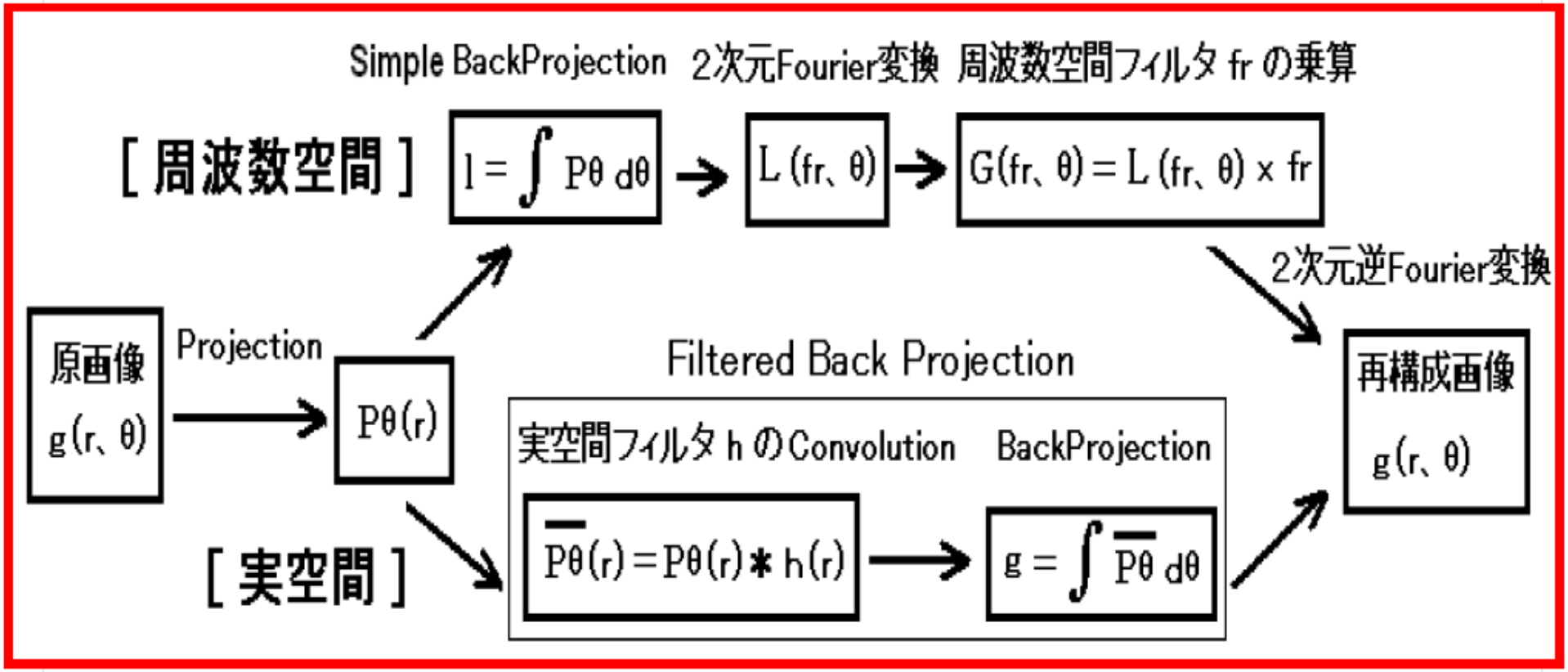
このテキストウィンドウ内をクリックする。
選択するプロジェクションデータは、フォルダ PETFBP 内の　PETsinogram を選択。

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 projectioを実行。
選択フィルタは２種類。
　RealRAMP256.txt
　RealSheppLogan256.txt

投影像 Pθ 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。

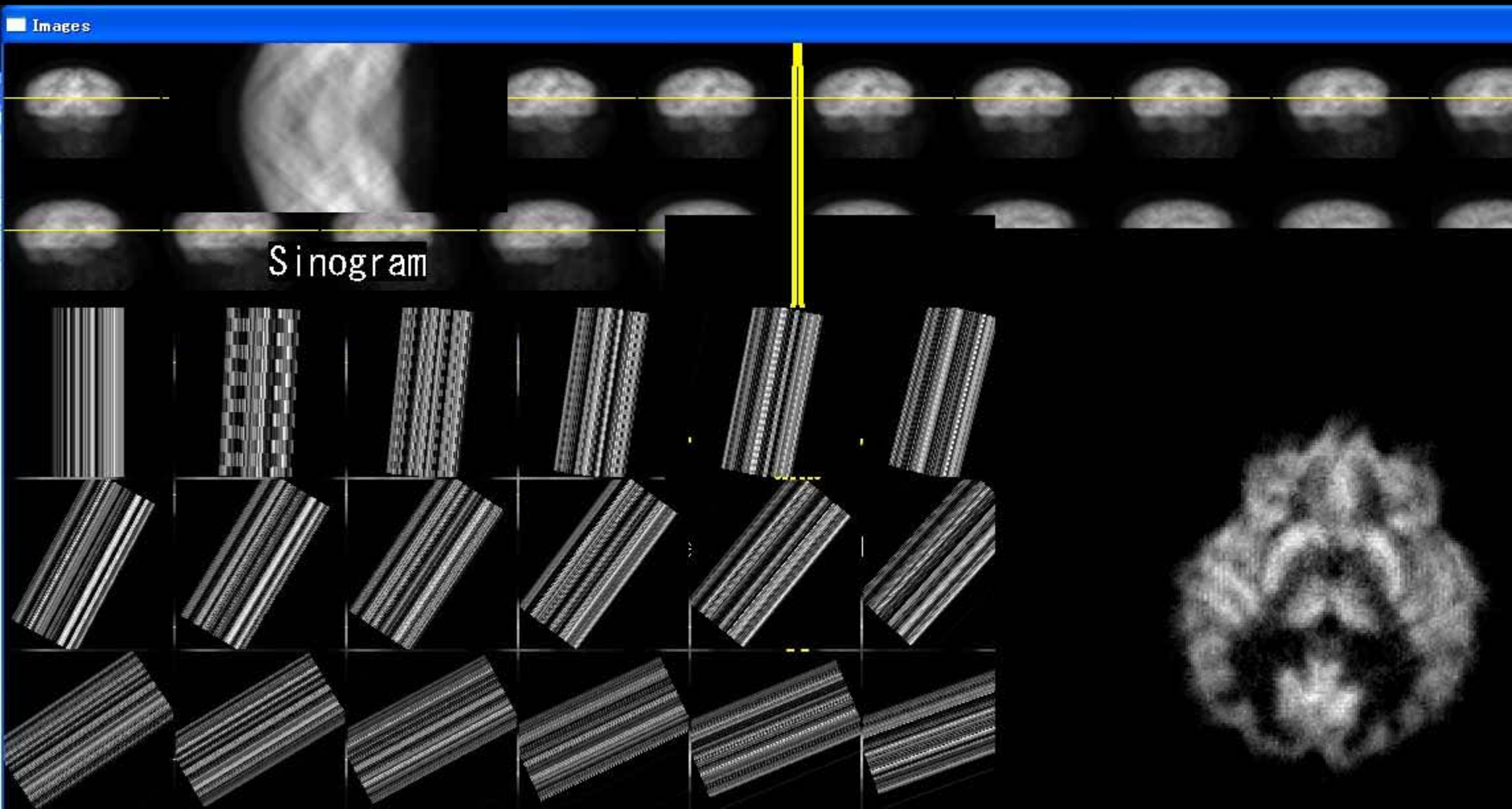
畳込みの定理を使うと、これらの周波数空間での処理が実空間で簡略化される。

投影像 Pθ 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。



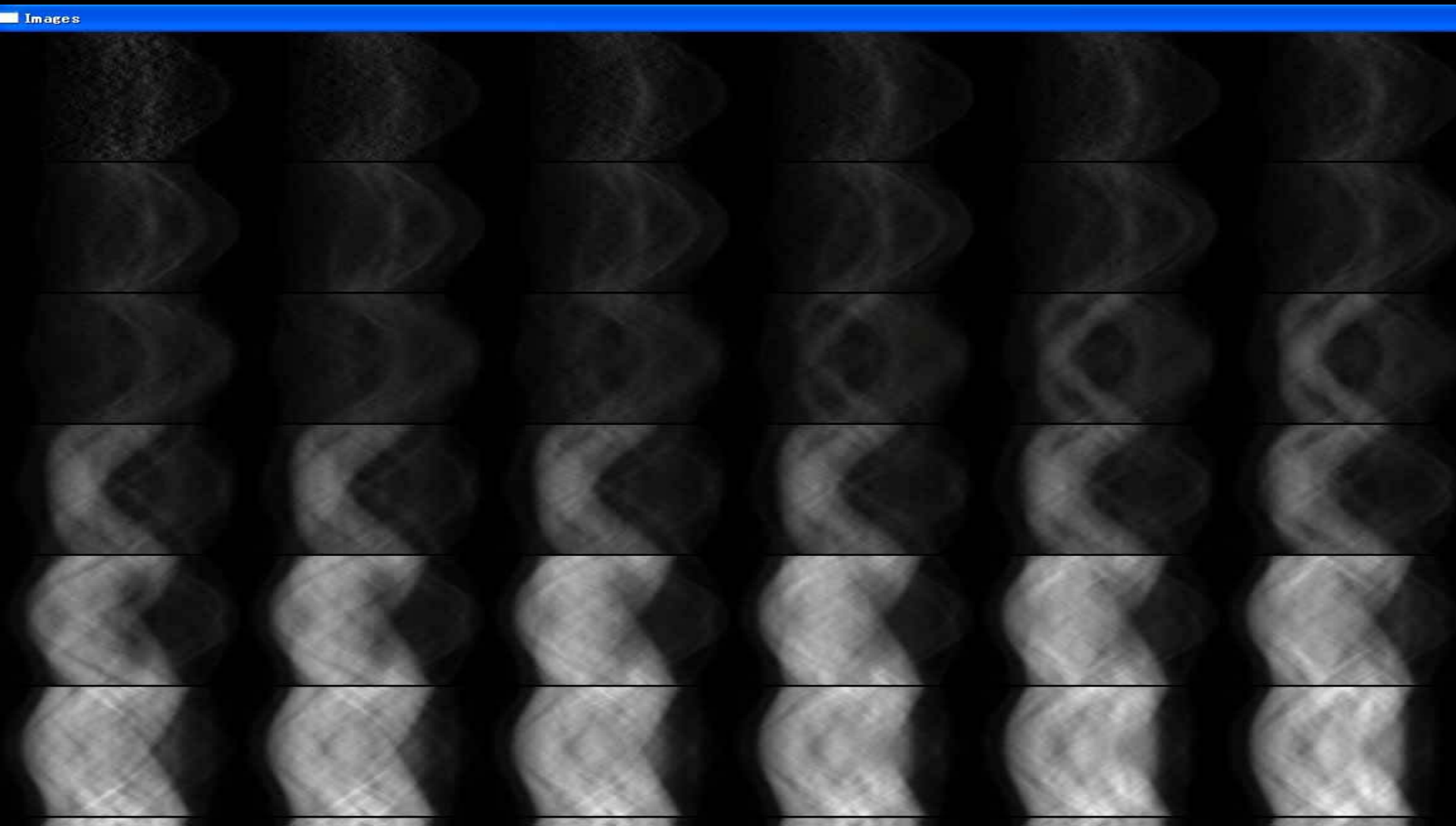
プログラム PETFBP.exe

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

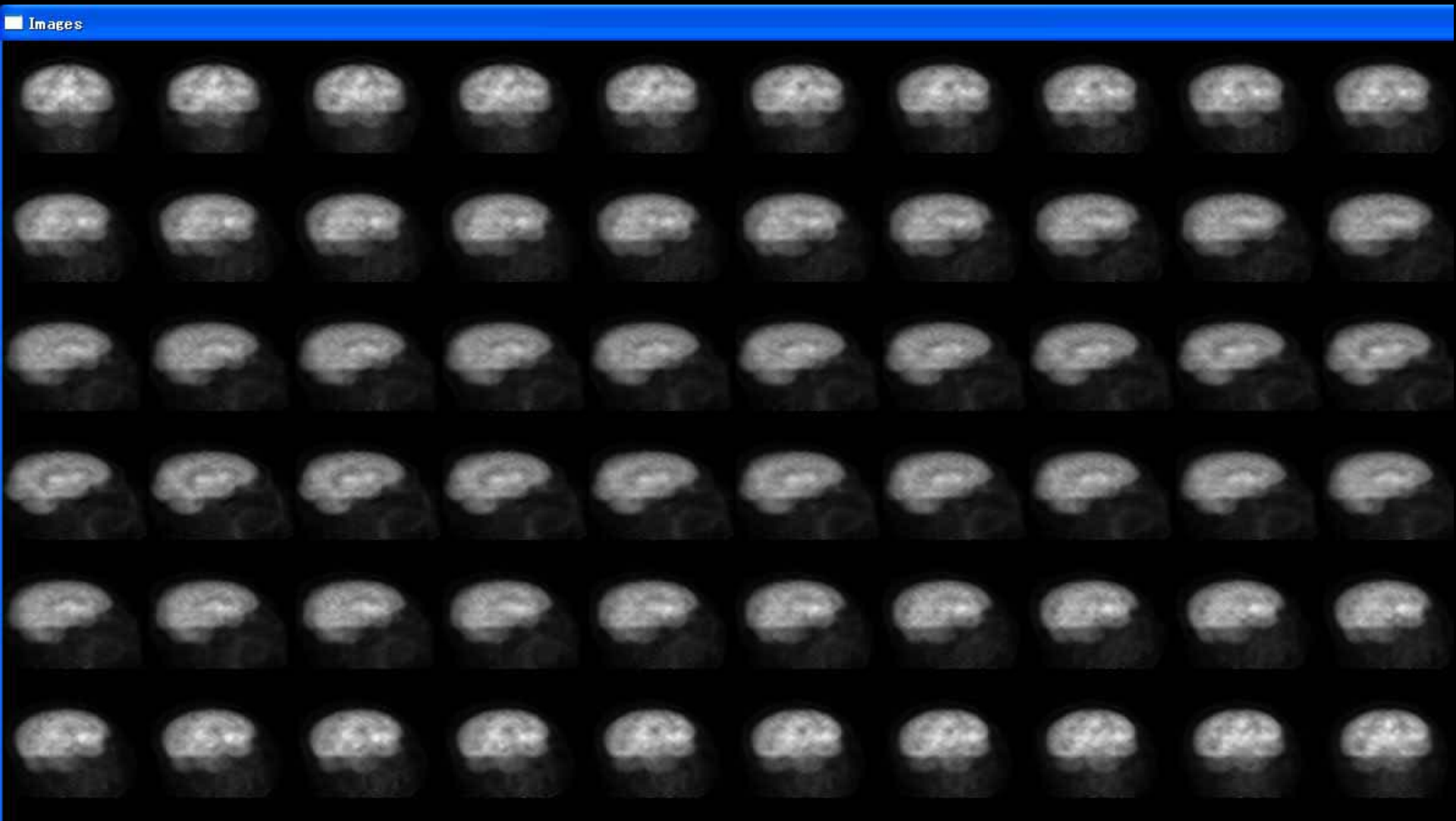


PETsinpgram ファイル

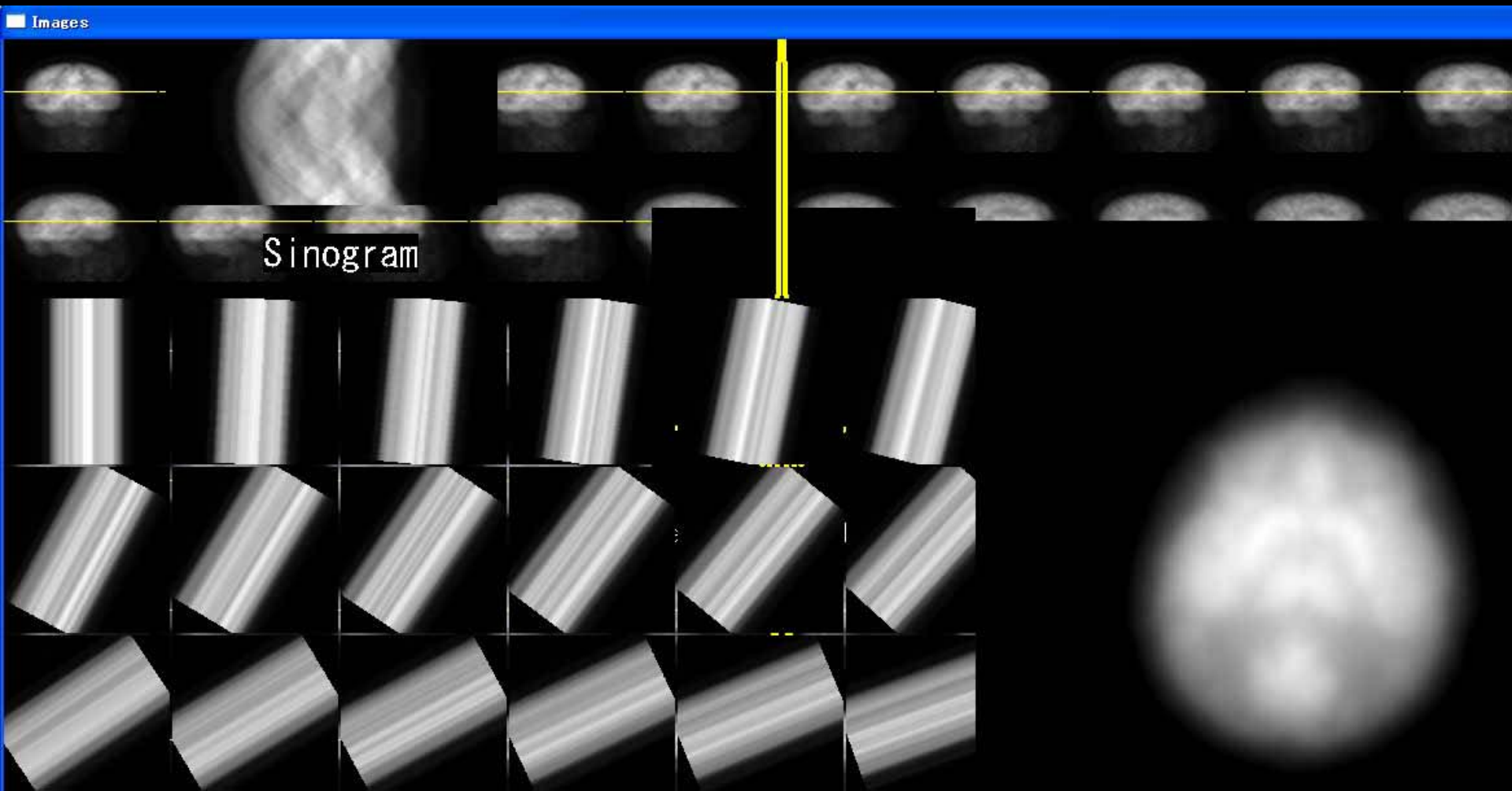
PETの収集データは各スライスのサイノグラムが並んでいる 3次元データ。(CTと同じ)。



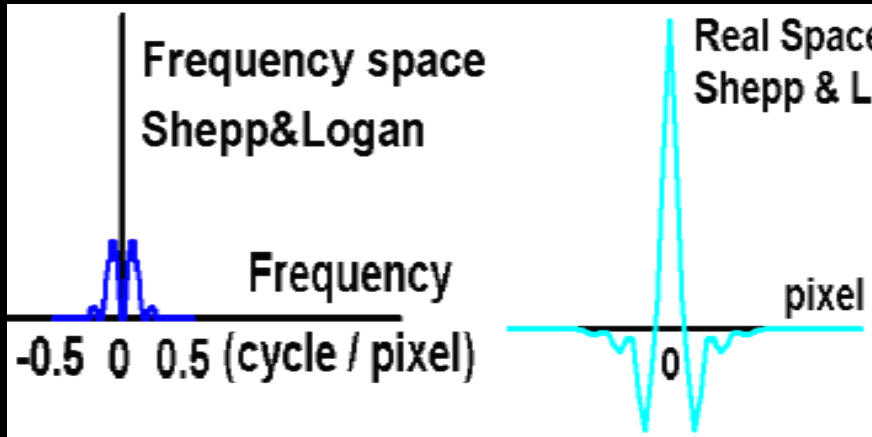
各スライスのサイノグラムが並んでいる3次元データを並べ替えれば、SPECTのプロジェクション像と同じ並びになる。 Sinogram[i][θ][j] = Projection [i][j][θ]



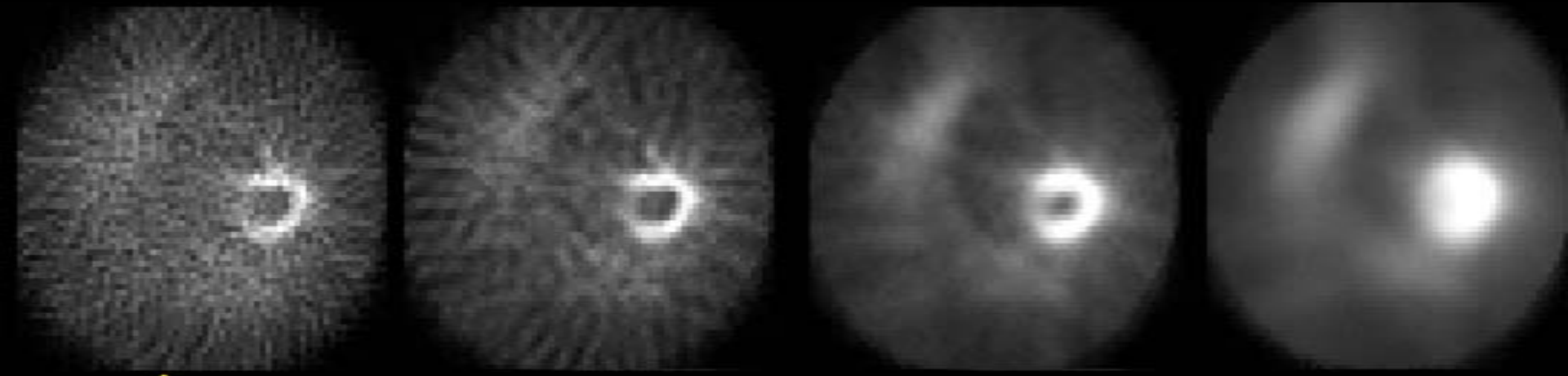
PETFBP.c を実行して、Simple BackProjection と
FBP の 再構成画像の違いを観察し、
Ramp等のフィルタ関数の必要性を理解してください。

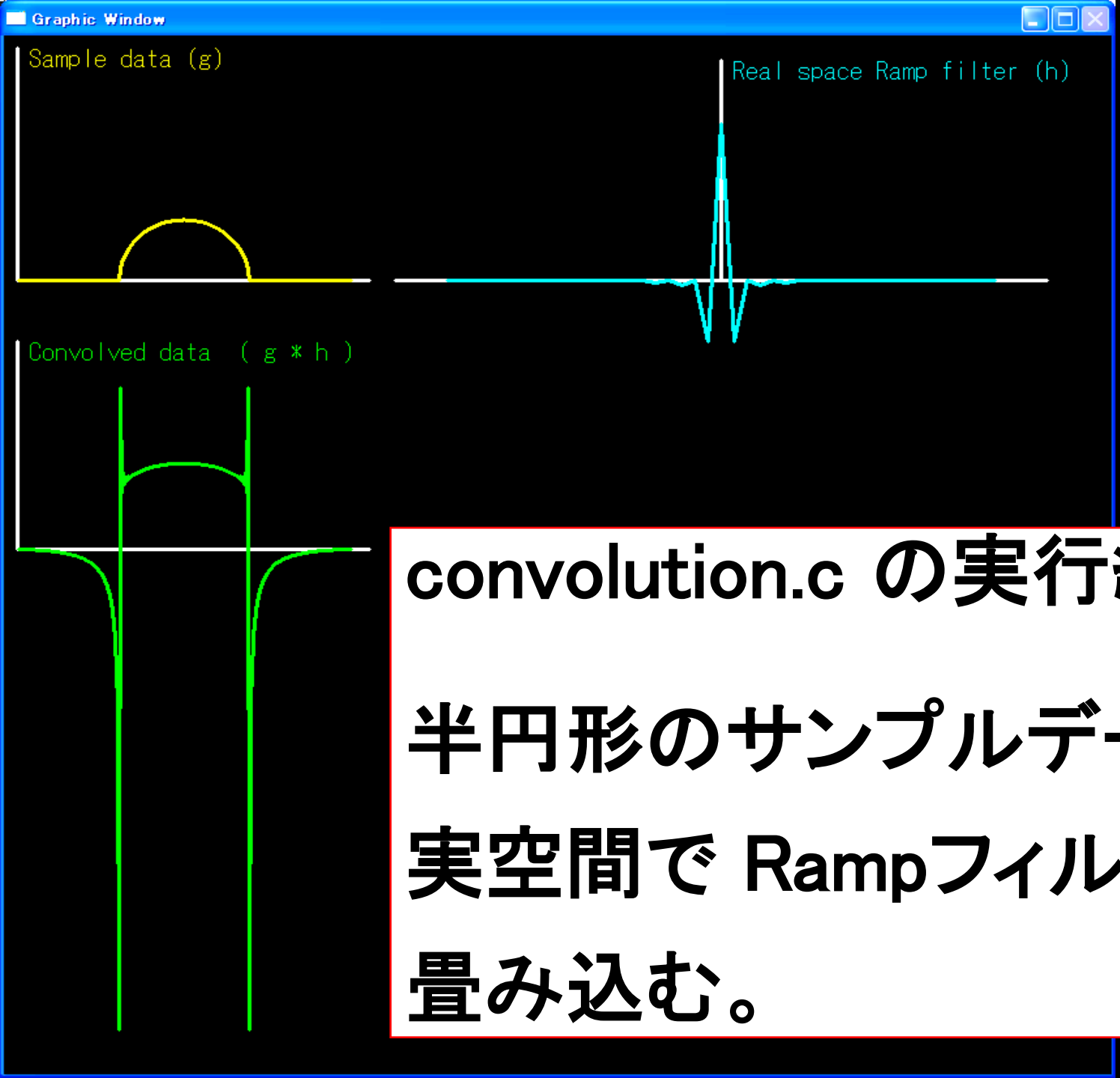


Rampフィルタは理論的には正確な再構成フィルタだが、
実際の臨床データに適用すると ナイキスト周波数以上の高周波成分を不連続に遮断するために生じる高周波ノイズや 放射状アーチファクトが目立つ場合が多いため、
臨床では高周波成分を抑制する工夫を施した再構成フィルタが用いられている。
SPECTで用いられる一般的な再構成フィルタとして、
Shepp＆Loganフィルタがある。
周波数空間上で、ナイキスト周波数近傍の高周波成分を連続的に減衰させるように設計されており
再構成画像に生じる**高周波ノイズや放射状アーチファクトを抑制**する効果をもつ。



実空間フィルタを畳込んだ2次元透視画像（$\overline{P\theta}$）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。





convolution.c の実行結果

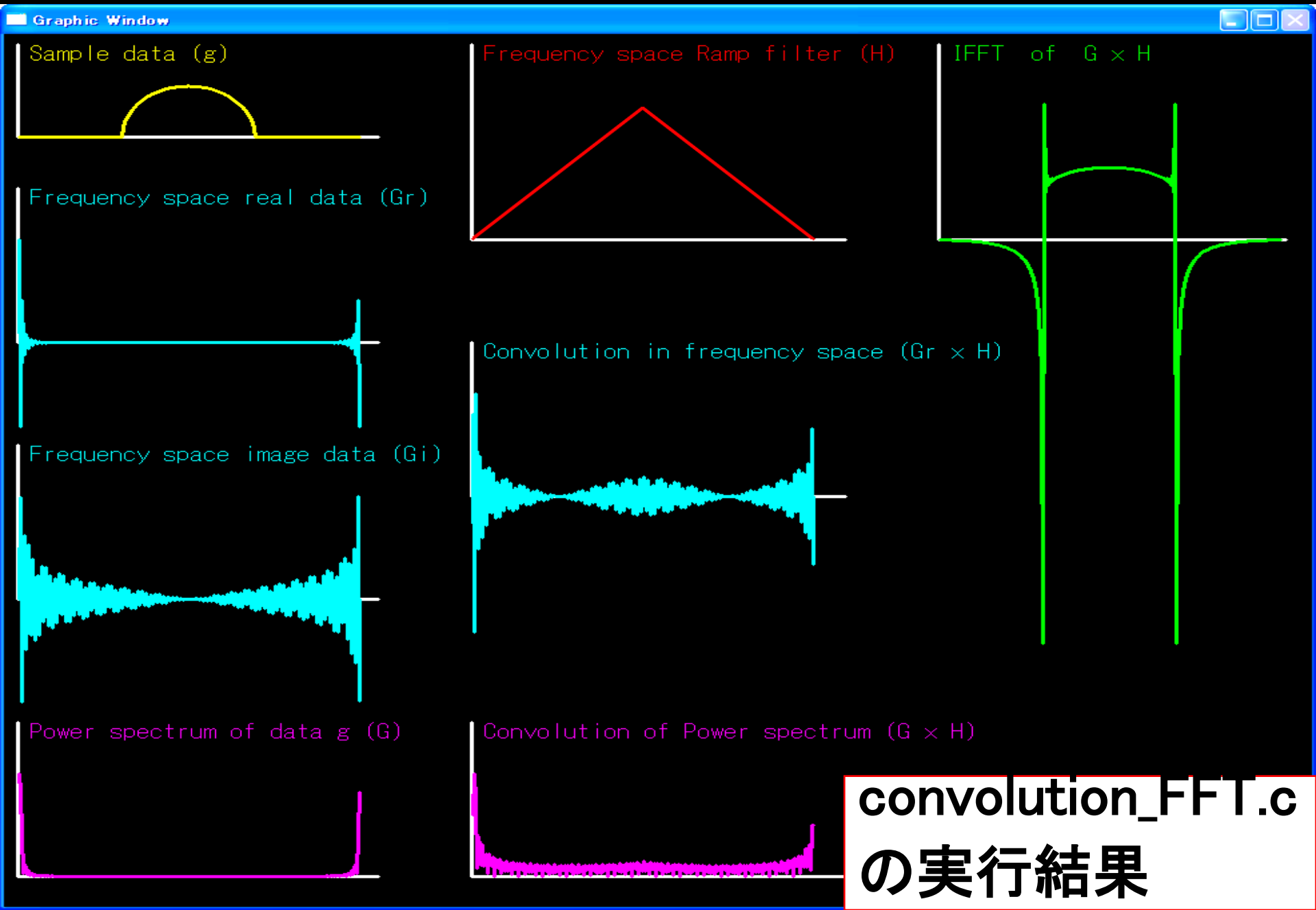
半円形のサンプルデータ g に
実空間で Rampフィルタ を
畳み込む。

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 ;
}
```

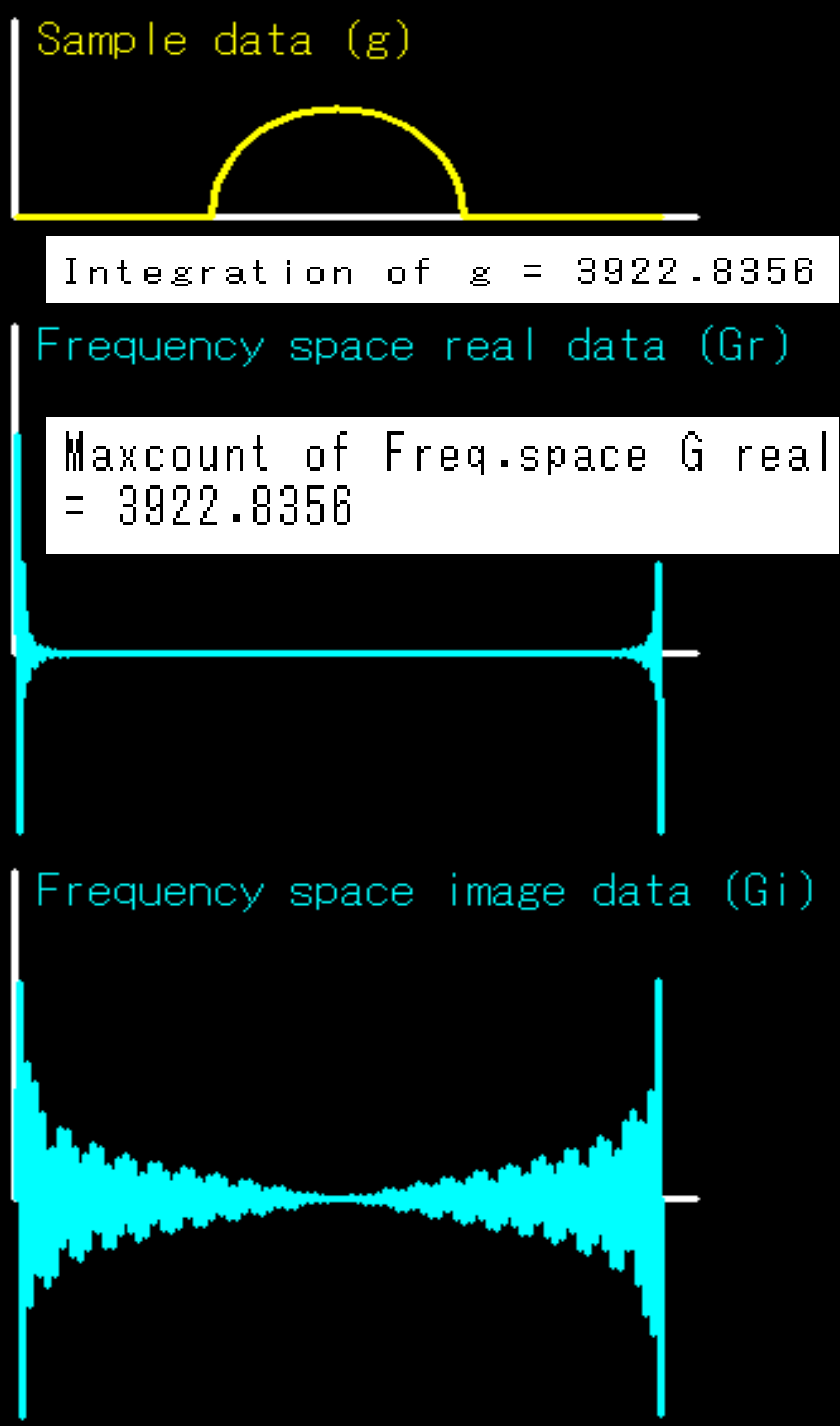
周波数空間で Rampフィルタ を かける



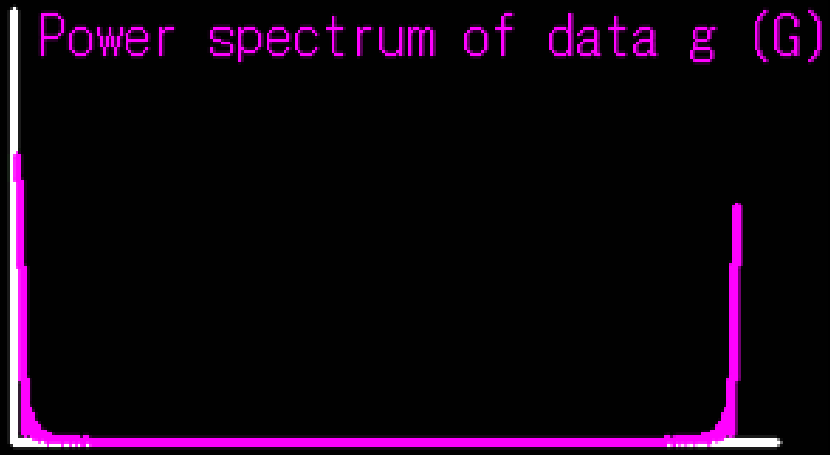
convolution_FFT.c
の実行結果

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。



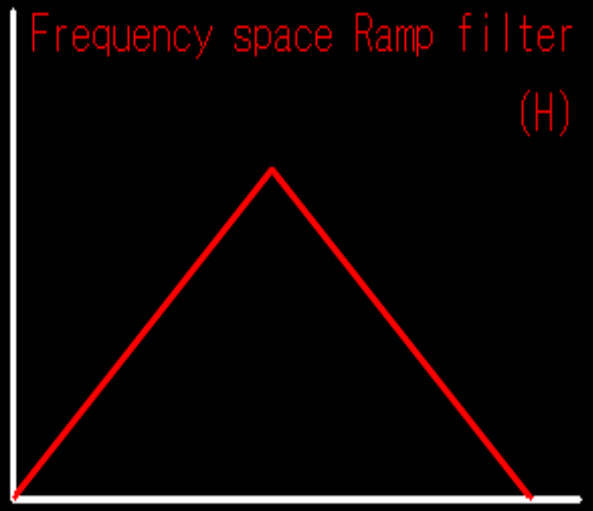
周波数空間の 実数成分 Gr、
虚数成分 Gi の二乗和の
平方根（パワースペクトル G ）
を表示。



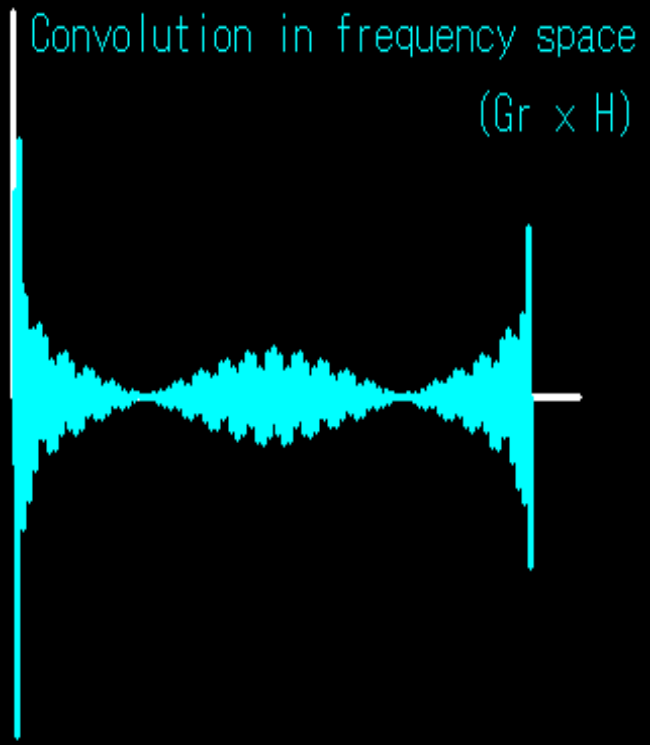
Maxcount of Power spectrum of G
= 3922.8356

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++ ){ GHi[i] = Gi[i] * H[i] ; }
```




```
//- Transform convolved GH into real space  -

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



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 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 x 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。）