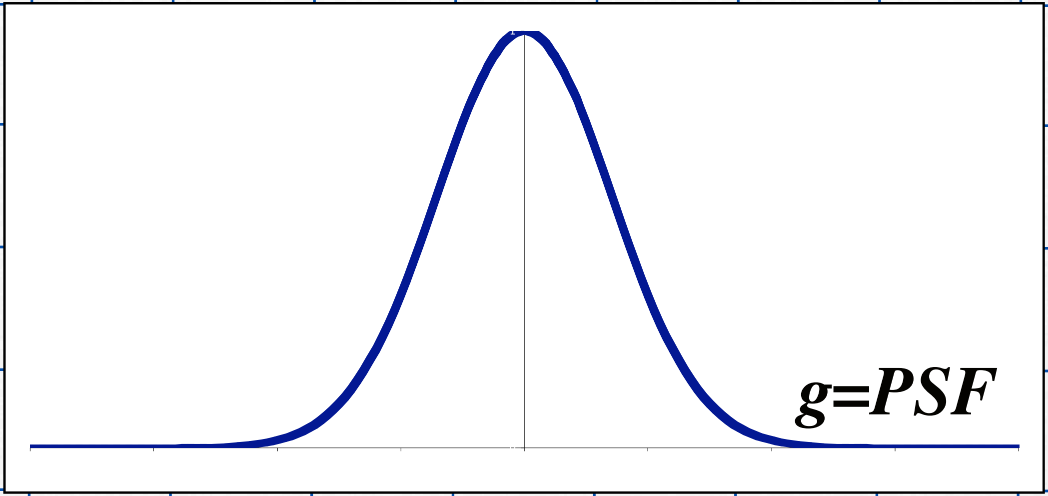
8th Plenary meeting and
HBM2004 Satellite meeting
Budapest June, 12, 2004

A new voxel-based PVE-correction method developed inside PVEOut project

Bruno Alfano

$$PET = C \otimes g$$



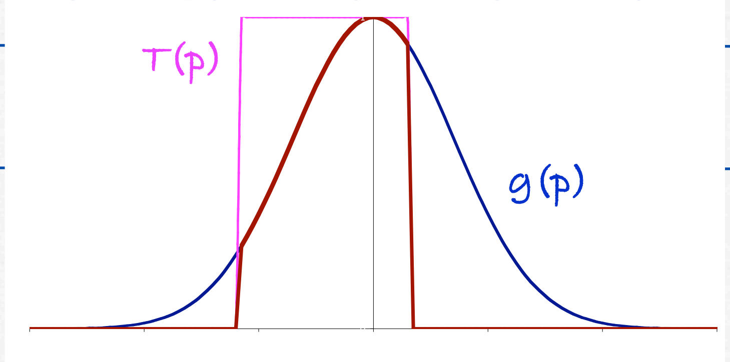
Each point of the radioactivity distribution function C becomes a spot with the shape of the Point Spread Function in the PET.

If C is the radioactivity distribution in the brain, the PET Count distribution is:

$$PET(p) = \int_{Volume} C(\rho)\cdot g(p-\rho)\cdot d\rho + noise$$

where g (= PSF of scanner) represents the probability function that a point C contributes to PET counts in its neighborhood.

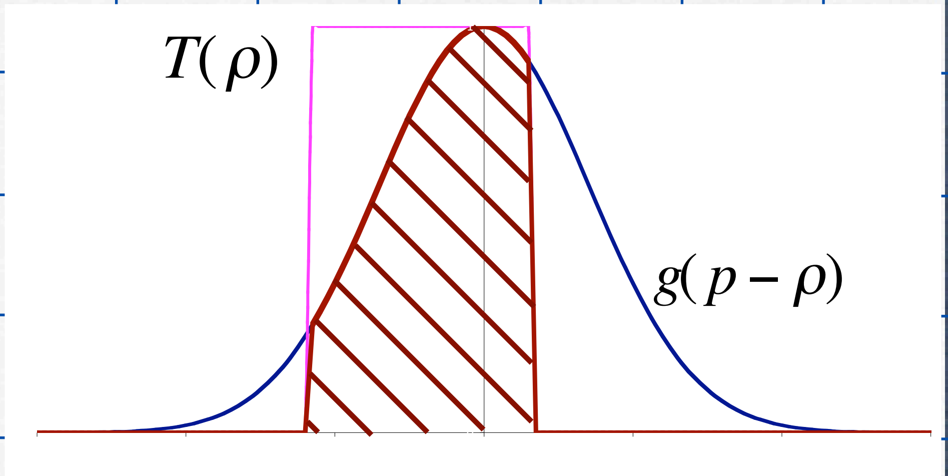
If we know where the tissue is (from segmentation), we have an information which improves the probability function. In fact, locally multiplying g(p) by the function T(p) (equal to 1 where there is tissue and zero otherwise), we obtain a more accurate probability function describing the possible sources of PET counts in p.



It can be defined as:

$$g'(p,\rho) = \frac{T(\rho)\cdot g(p-\rho)}{\int_{Volume} T(\psi)\cdot g(p-\psi)\cdot d\psi}$$

If the denominator is ≠ 0; $g'(p,\rho) = 0$ otherwise.



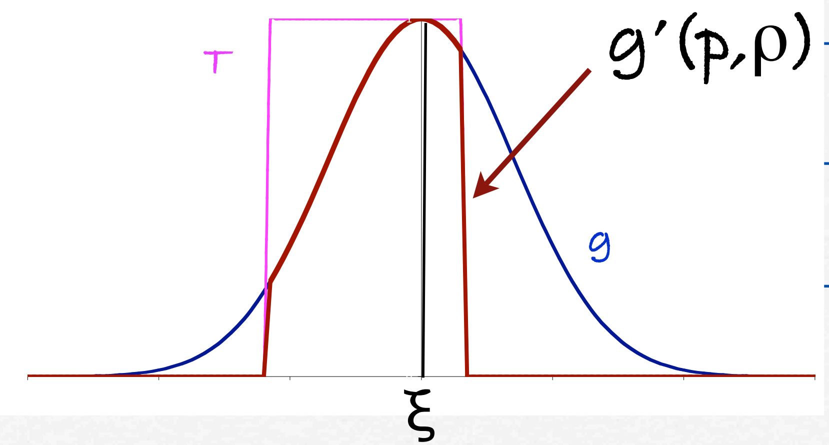
It represent the probability that a PET count in a point p come from the point ρ.

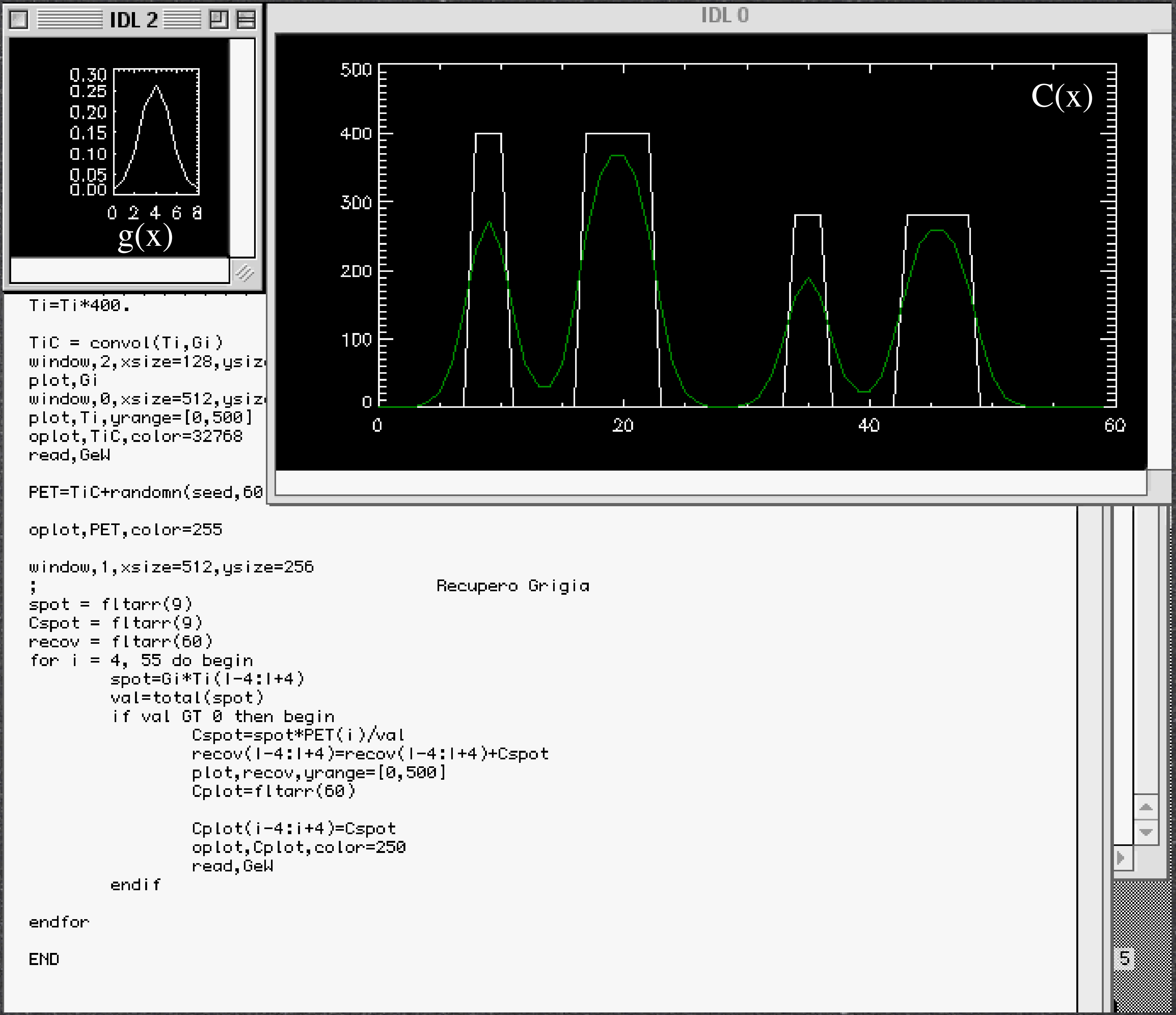
The PVE corrected (recovered) Grey Matter can be calculated as:

$$R_{GMPET}(p) = \int_{Volume} GMPET(\xi) \cdot g'(p,\xi) \cdot d\xi$$

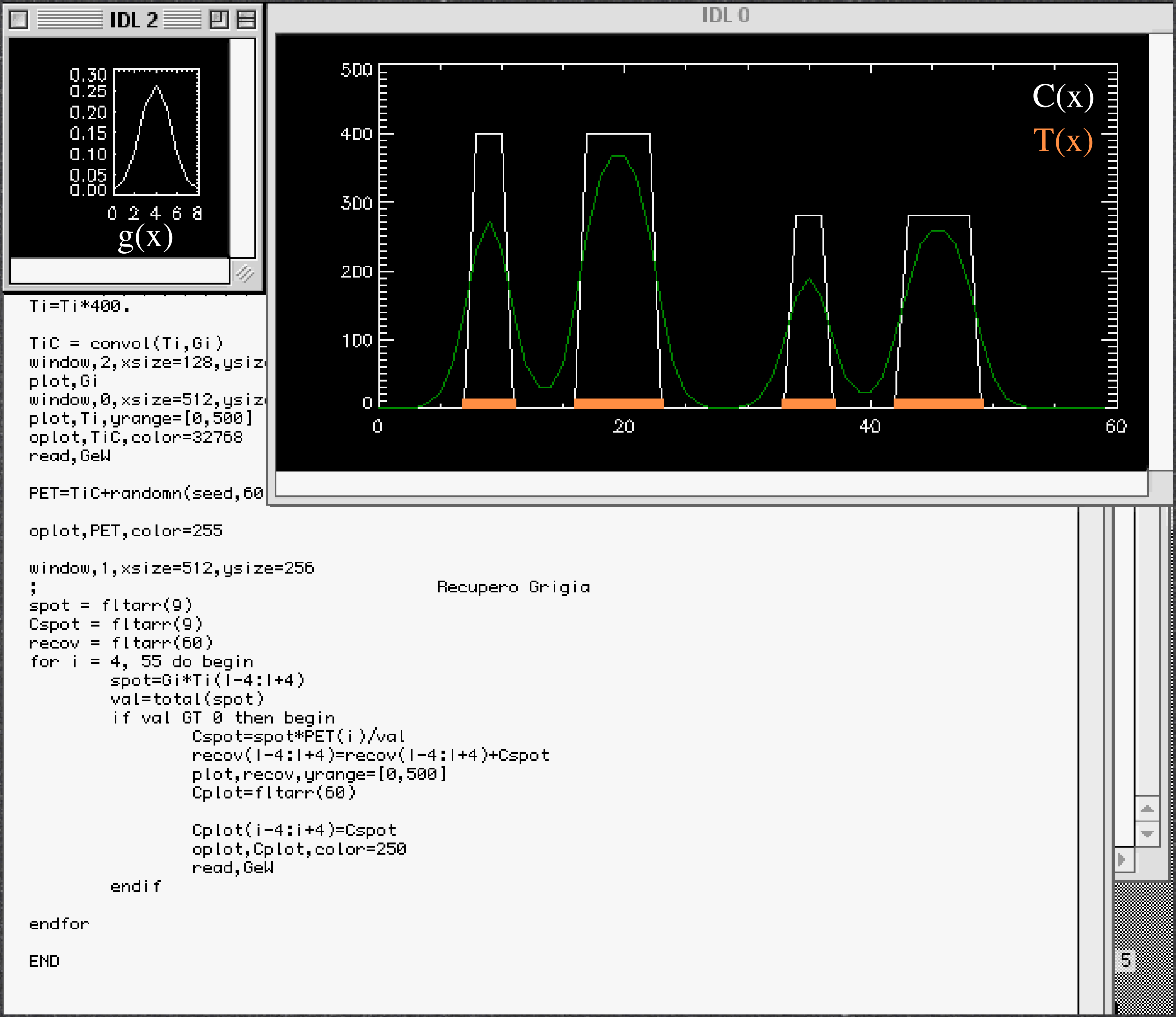
where GMPET represents the distribution of PET counts due to GM alone.

In other words, to reconstruct the activity concentration function C, the proposed algoritm spreads each point ξ of the function PET according to the probability function $g'(p,\rho)$, calculated in the neighbourhood of ξ.

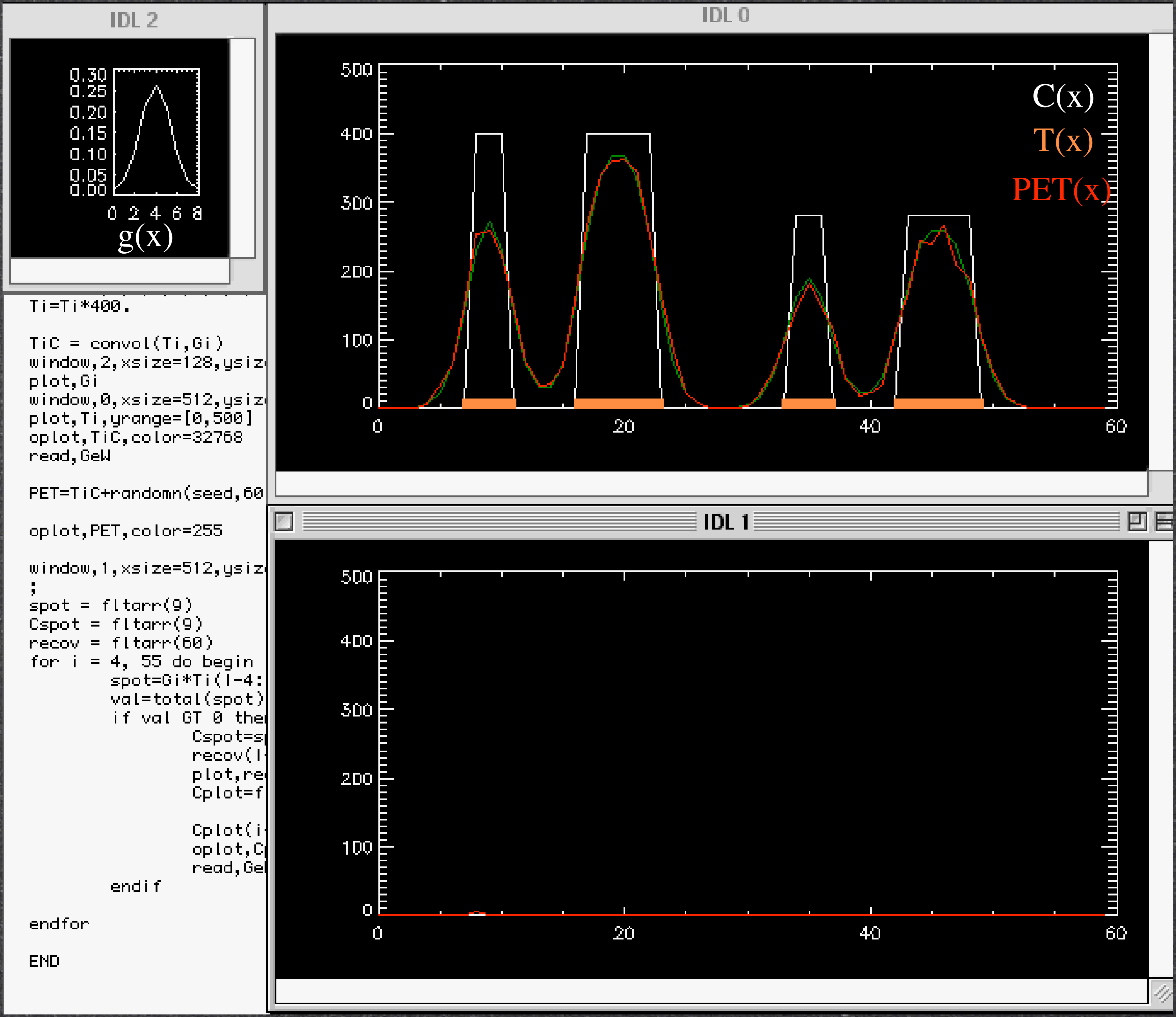




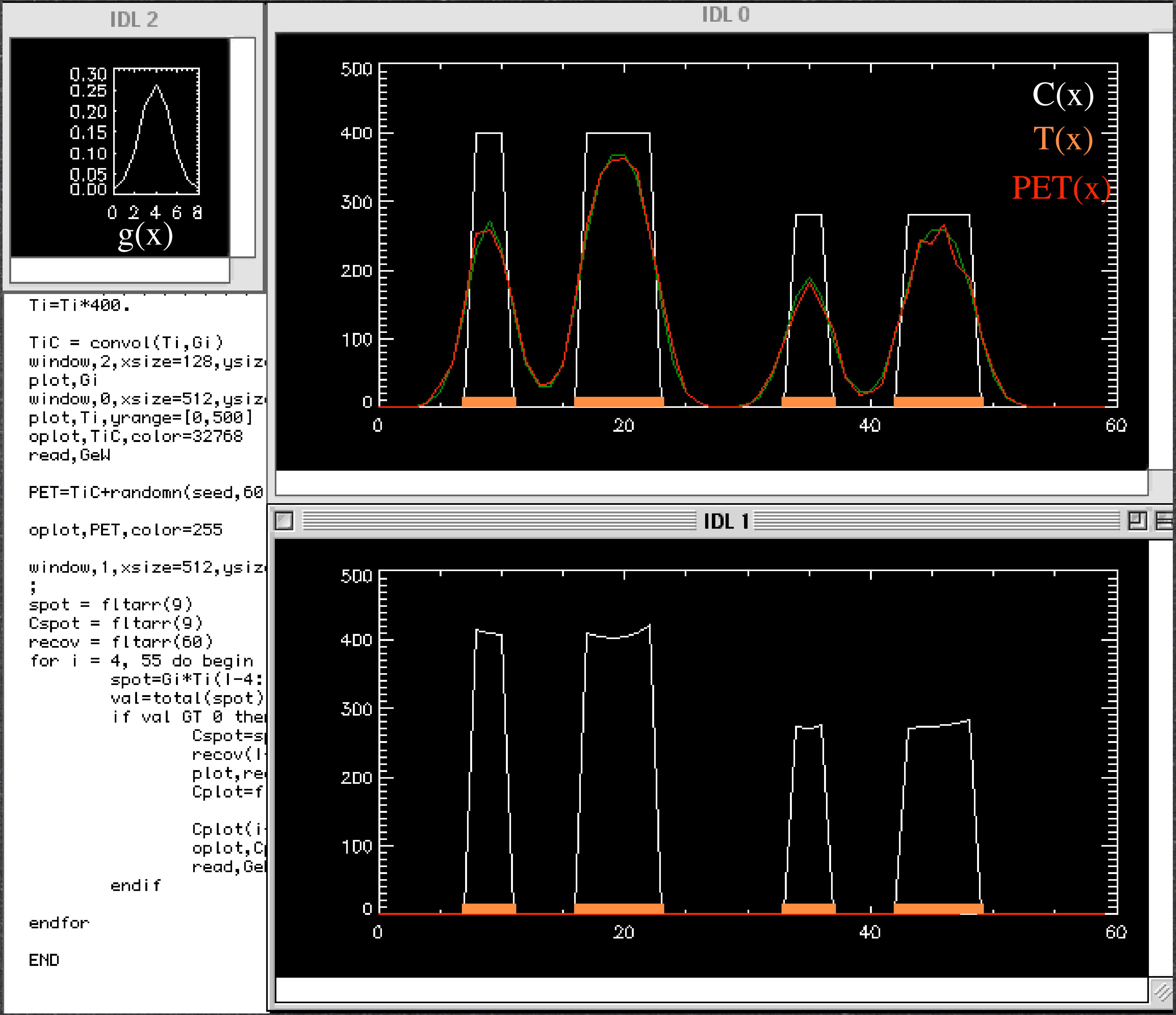
```
Ti=Ti*400.

TiC = convol(Ti,Gi)
window,2,xsize=128,ysiz
plot,Gi
window,0,xsize=512,ysiz
plot,Ti,yrange=[0,500]
oplot,TiC,color=32768
read,GeW

PET=TiC+randomn(seed,60

oplot,PET,color=255

window,1,xsize=512,ysize=256
;                                       Recupero Grigia
spot = fltarr(9)
Cspot = fltarr(9)
recov = fltarr(60)
for i = 4, 55 do begin
        spot=Gi*Ti(I-4:I+4)
        val=total(spot)
        if val GT 0 then begin
                Cspot=spot*PET(i)/val
                recov(I-4:I+4)=recov(I-4:I+4)+Cspot
                plot,recov,yrange=[0,500]
                Cplot=fltarr(60)

                Cplot(i-4:i+4)=Cspot
                oplot,Cplot,color=250
                read,GeW
        endif

endfor

END
```



```
Ti=Ti*400.

TiC = convol(Ti,Gi)
window,2,xsize=128,ysiz
plot,Gi
window,0,xsize=512,ysiz
plot,Ti,yrange=[0,500]
oplot,TiC,color=32768
read,GeW

PET=TiC+randomn(seed,60

oplot,PET,color=255

window,1,xsize=512,ysize=256
;                                       Recupero Grigia
spot = fltarr(9)
Cspot = fltarr(9)
recov = fltarr(60)
for i = 4, 55 do begin
        spot=Gi*Ti(I-4:I+4)
        val=total(spot)
        if val GT 0 then begin
                Cspot=spot*PET(i)/val
                recov(I-4:I+4)=recov(I-4:I+4)+Cspot
                plot,recov,yrange=[0,500]
                Cplot=fltarr(60)

                Cplot(i-4:i+4)=Cspot
                oplot,Cplot,color=250
                read,GeW
        endif

endfor

END
```



```
Ti=Ti*400.

TiC = convol(Ti,Gi)
window,2,xsize=128,ysiz
plot,Gi
window,0,xsize=512,ysiz
plot,Ti,yrange=[0,500]
oplot,TiC,color=32768
read,GeW

PET=TiC+randomn(seed,60

oplot,PET,color=255

window,1,xsize=512,ysiz
;
spot = fltarr(9)
Cspot = fltarr(9)
recov = fltarr(60)
for i = 4, 55 do begin
        spot=Gi*Ti(I-4:
        val=total(spot)
        if val GT 0 the
                Cspot=s
                recov(I
                plot,re
                Cplot=f

                Cplot(i
                oplot,C
                read,Ge
        endif

endfor

END
```

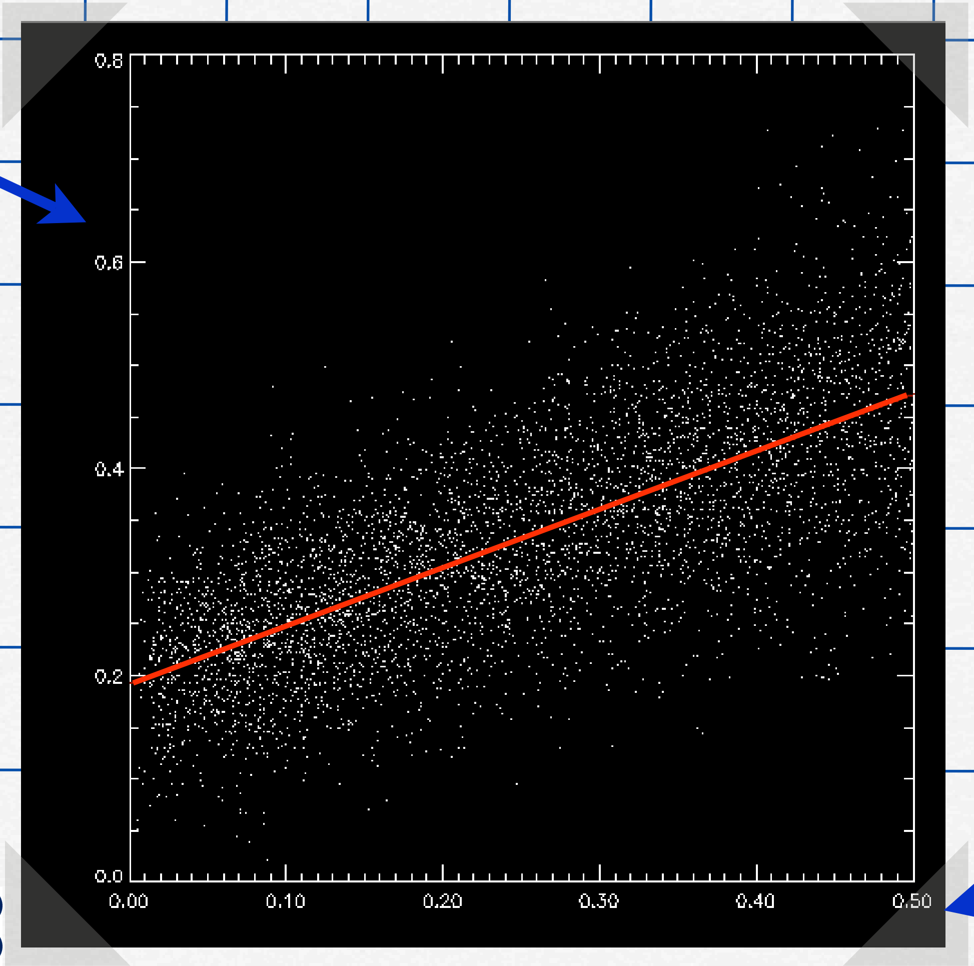


```
Ti=Ti*400.

TiC = convol(Ti,Gi)
window,2,xsize=128,ysiz
plot,Gi
window,0,xsize=512,ysiz
plot,Ti,yrange=[0,500]
oplot,TiC,color=32768
read,GeW

PET=TiC+randomn(seed,60

oplot,PET,color=255

window,1,xsize=512,ysiz
;
spot = fltarr(9)
Cspot = fltarr(9)
recov = fltarr(60)
for i = 4, 55 do begin
        spot=Gi*Ti(l-4:
        val=total(spot)
        if val GT 0 the
                Cspot=s
                recov(l
                plot,re
                Cplot=f

                Cplot(i
                oplot,C
                read,Ge
        endif

endfor

END
```

Since the algoritm can operate each time only on one tissue, to apply the method we need:

- ☐ Segmented Grey Matter
- ☐ Segmented White Matter
- ☐ Registered PET
- ☐ The Point Spread Function (PSF) of the PET
- ☐ A Method for separating GM-PET from WM-PET

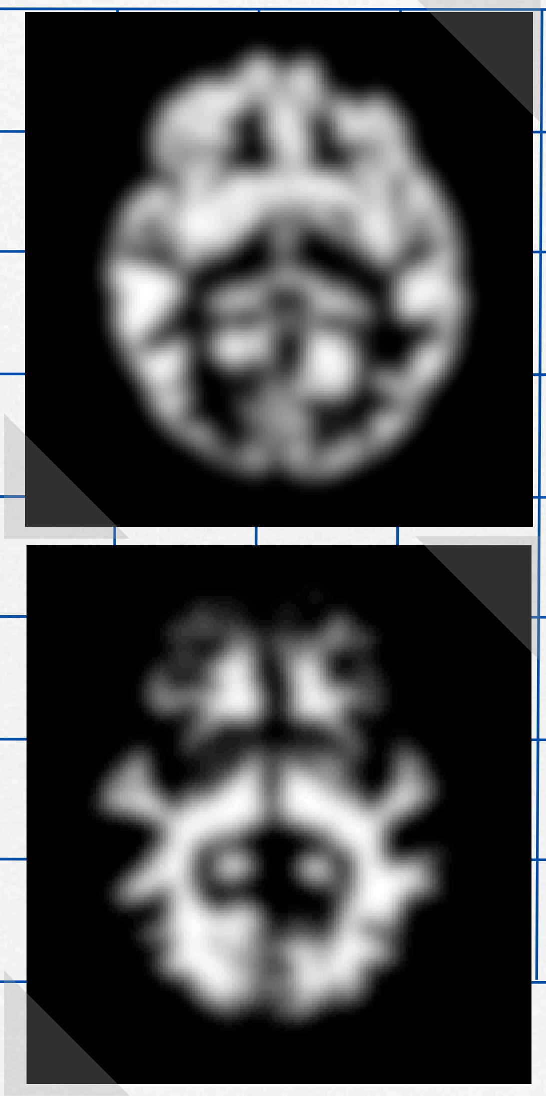
WM activity concentration can be estimated:



1) in Centri semiovali, as proposed by Müller-Gartner;
2) using the value calculated with Rousset method;
3) from a fitting

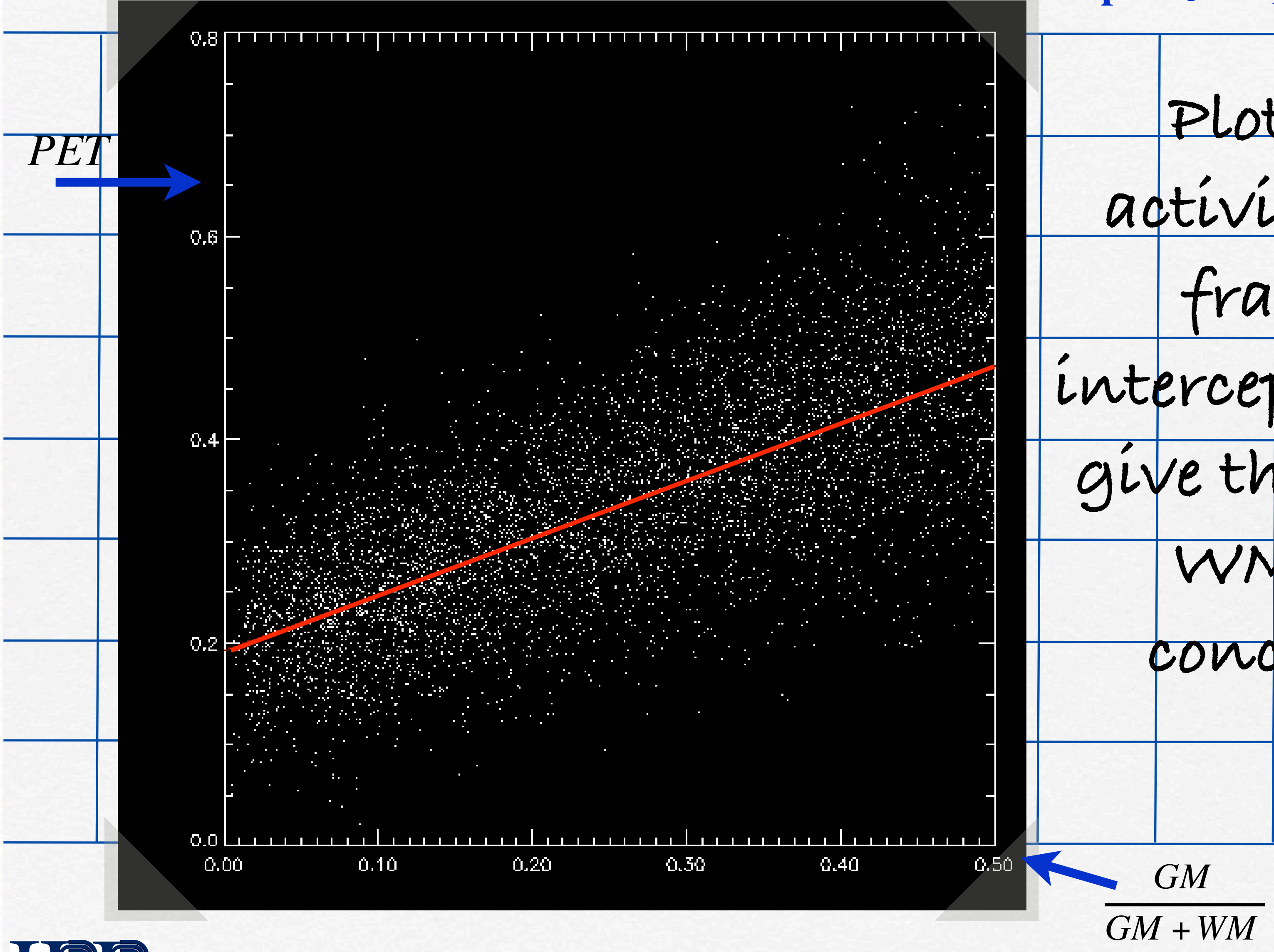
Convoluting the segmented GM and the segmented WM with the PSF of the PET, we can evaluate voxel by voxel the ratio

$$\frac{GM}{GM + WM}$$



PVEOut

8th Plenary meeting and HBM2004 Satellite meeting
Budapest June, 12, 2004

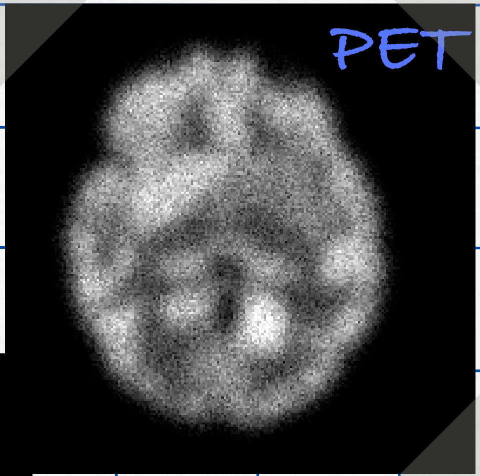


$$\frac{GM}{GM+WM}$$

Plotting PET activity VS. GM fraction, the intercept of a fitting give the estimate of WM activity concentration.

IBB

At last, subtracting the scaled convoluted WM from the PET, we separate the contribution of the two tissues to the PET and can apply the PVE correction method separately to GM and WM.

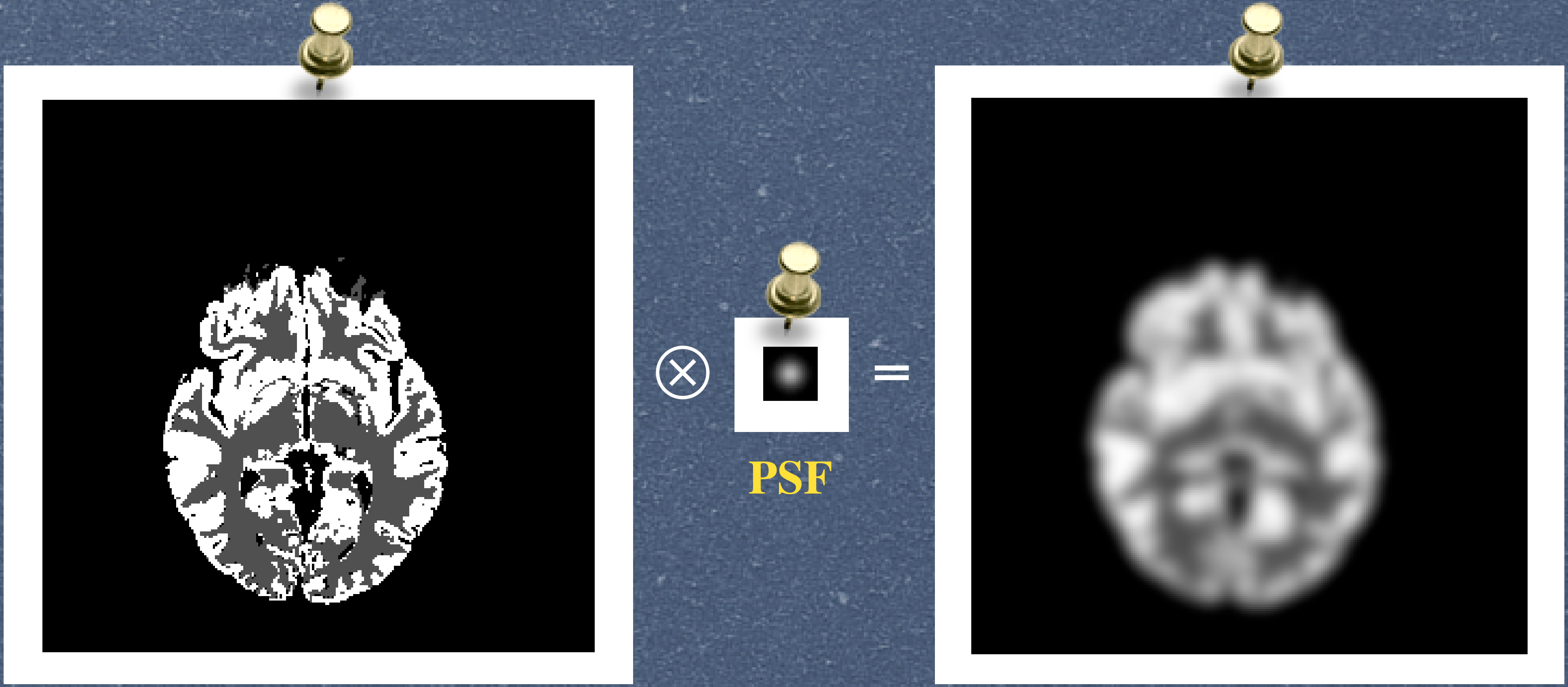

PET


PET of WM

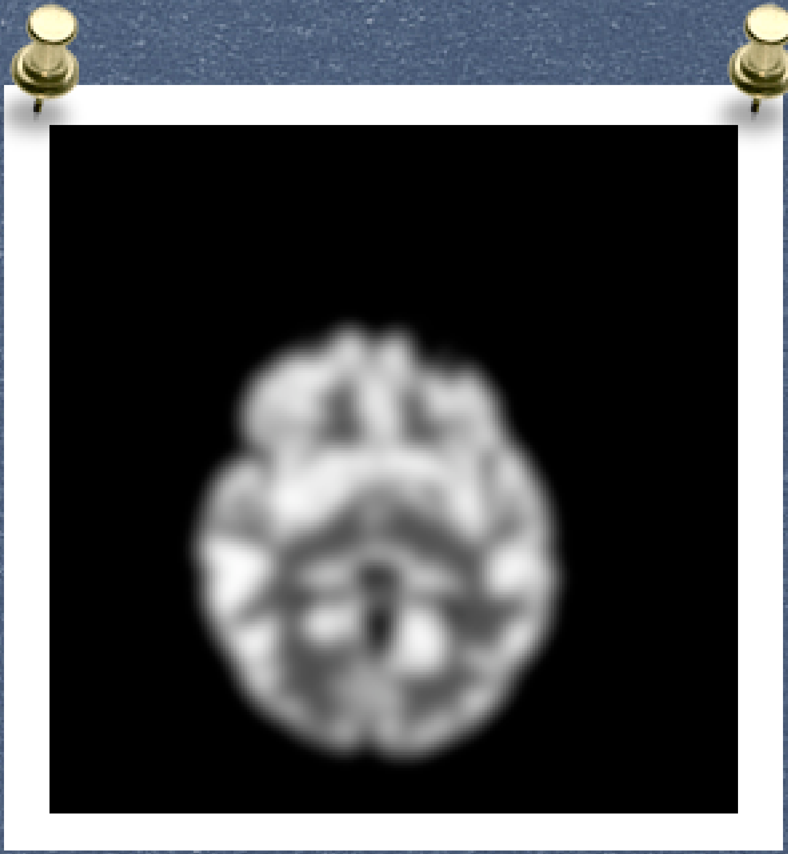

PET of GM

Computer-built phantom

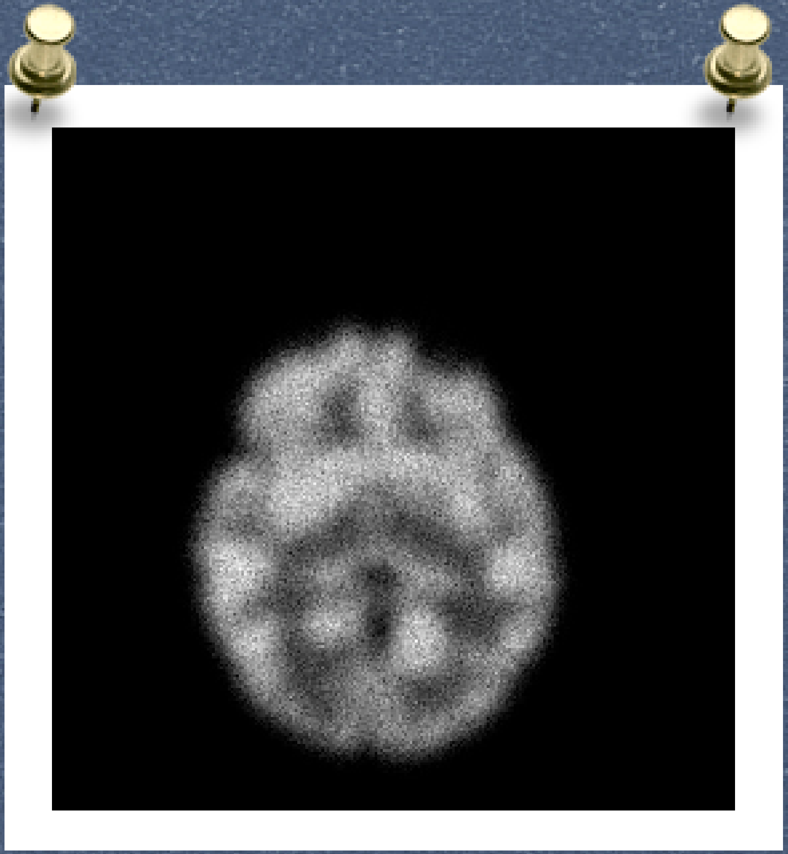
Computer-built phantom



Computer-built phantom



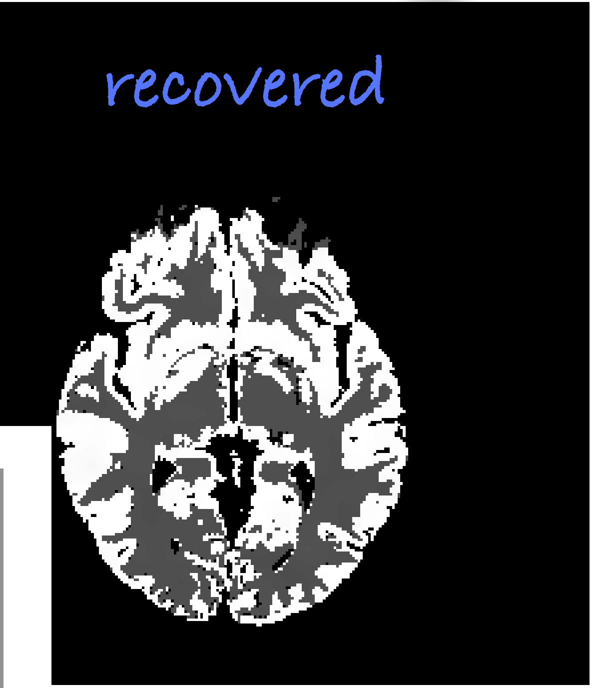
+ noise =



Materials

Results

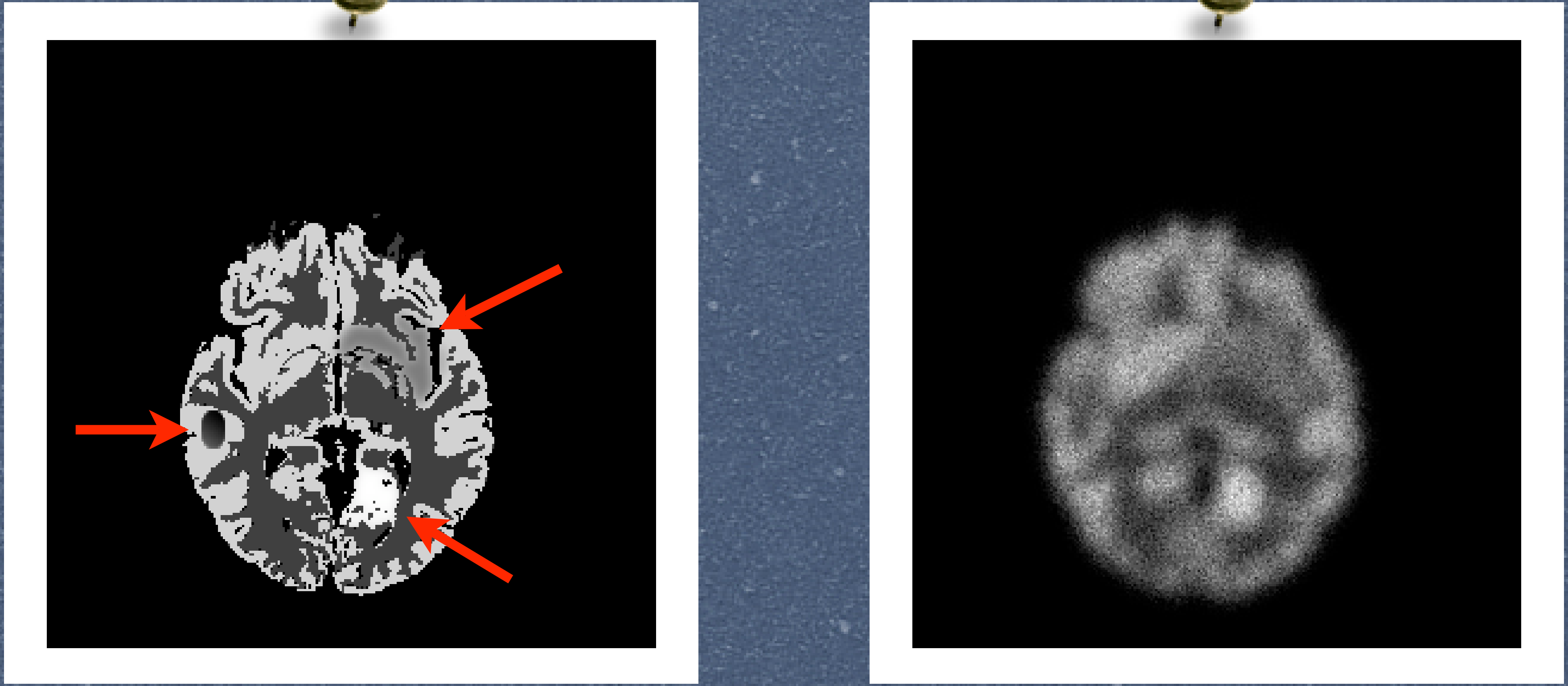
original



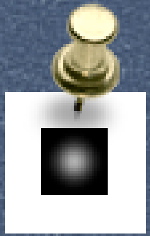
recovered

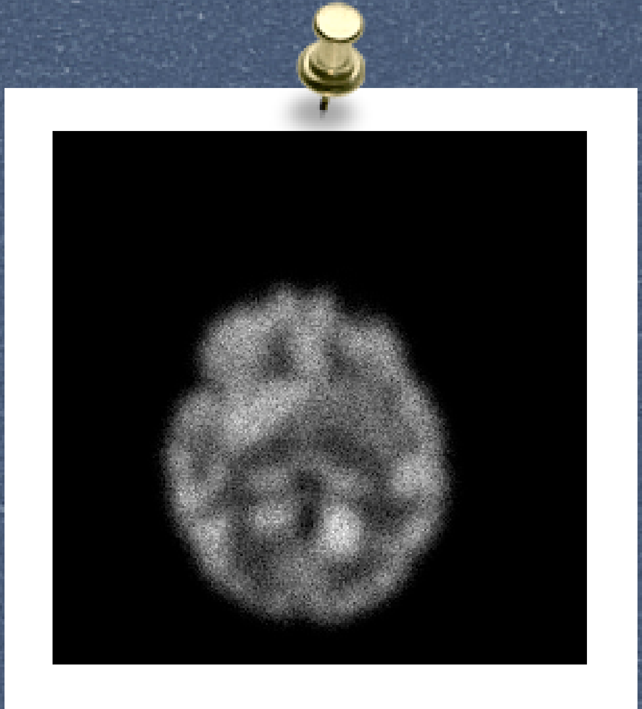
error x 2

Computer-built phantom with “lesions”

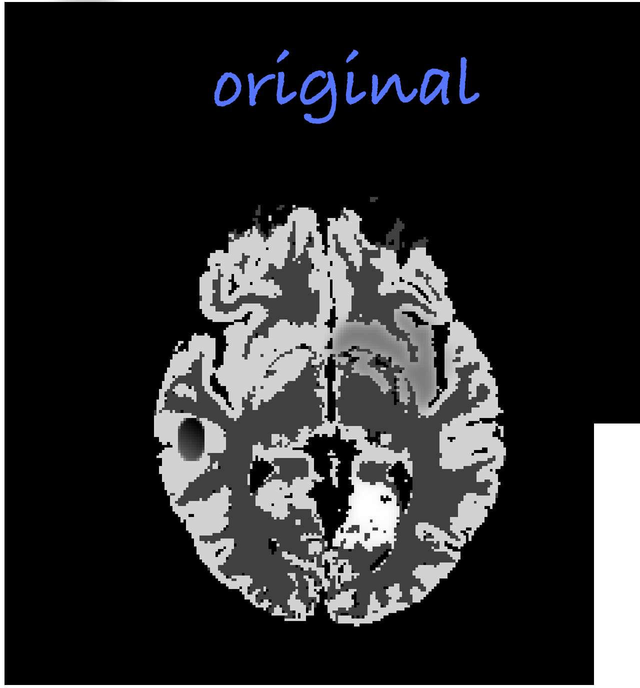


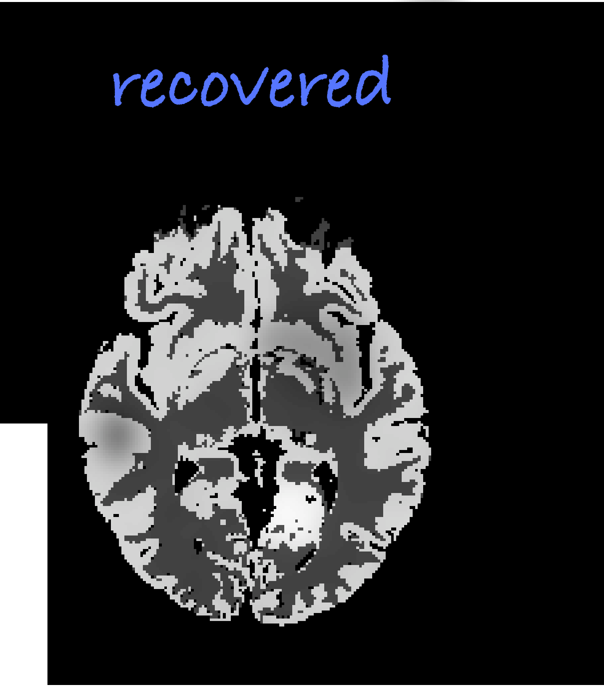
Materials



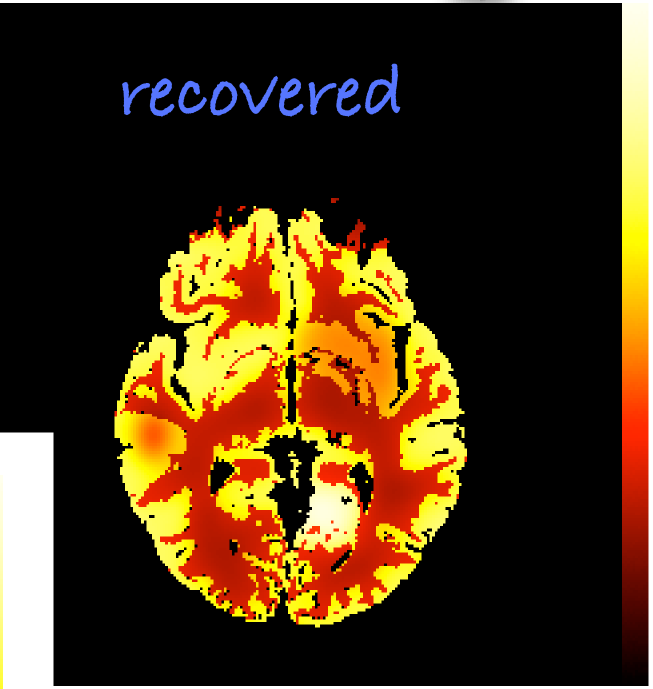


Results





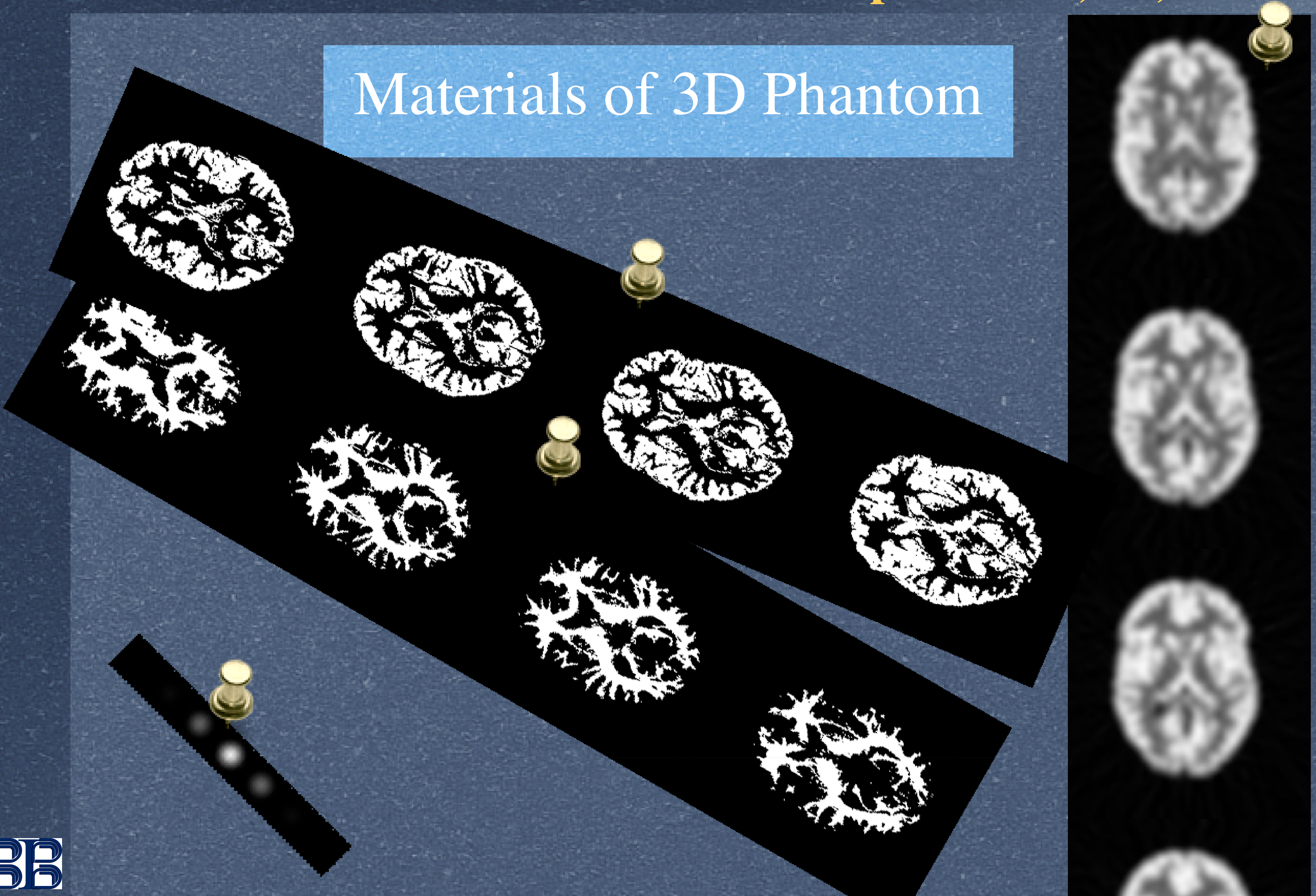
Results



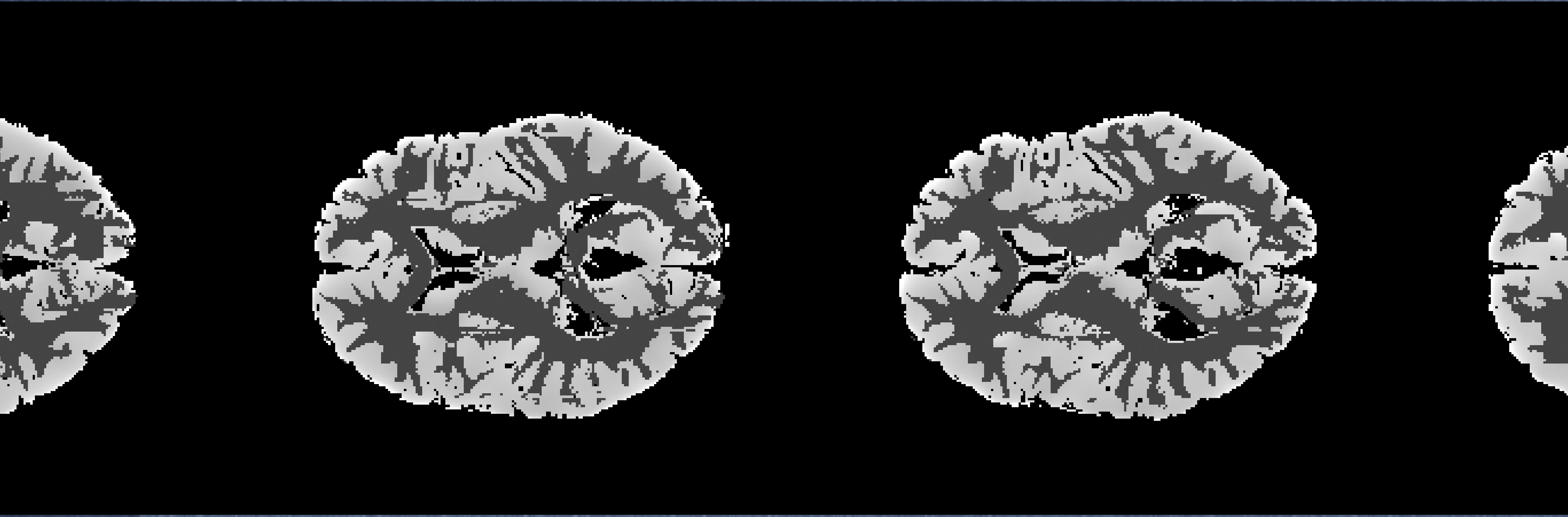
PVEOut

8th Plenary meeting
Budapest June, 12, 2004

Materials of 3D Phantom



Results

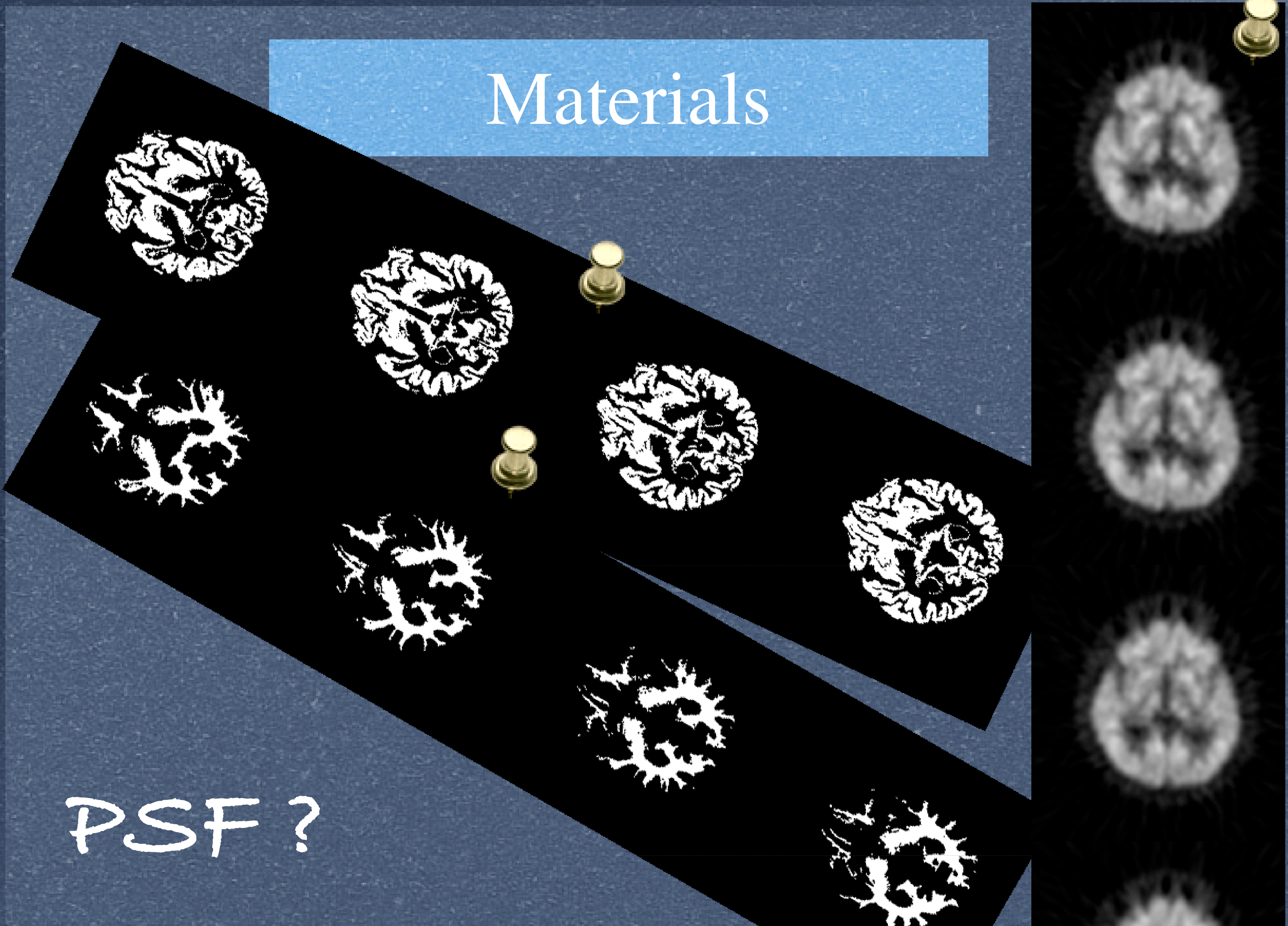


Original values:
GM=255 WM=64

Mean recovered values:
GM=254.69 WM=65.67

Real subject

Materials

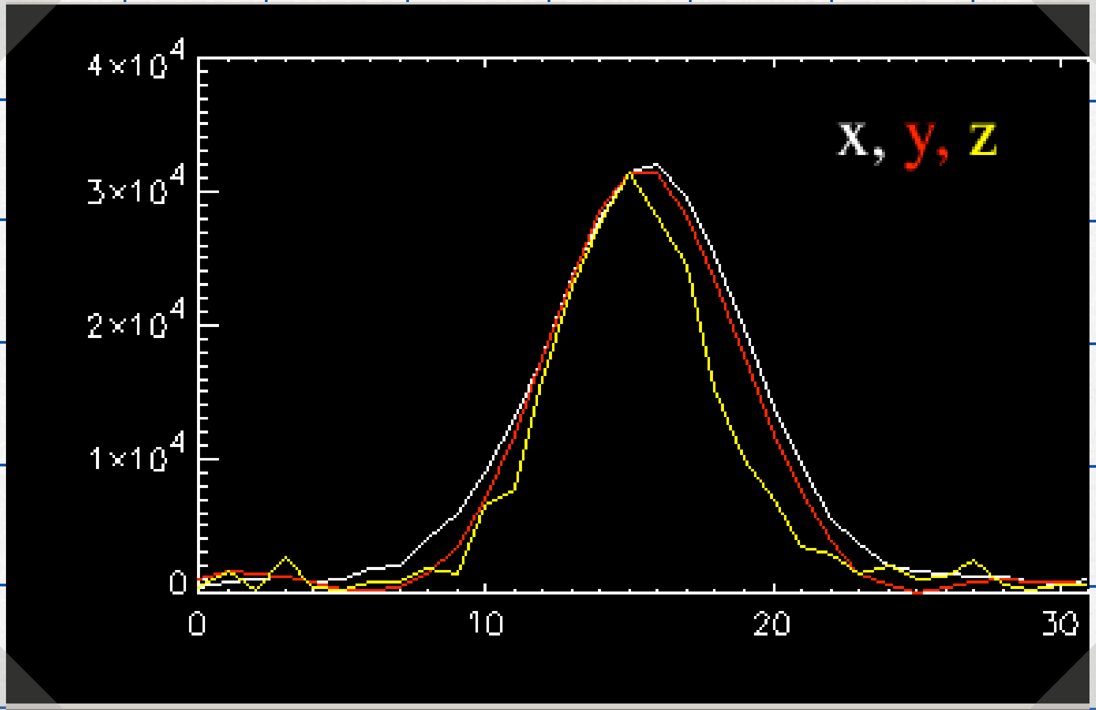


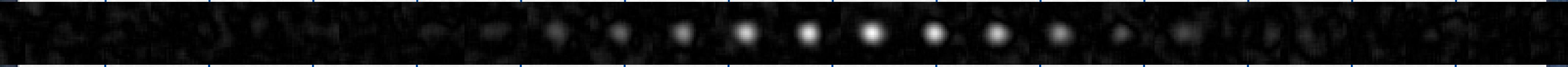
PSF ?

$$PET \cong C \otimes PSF + noise$$

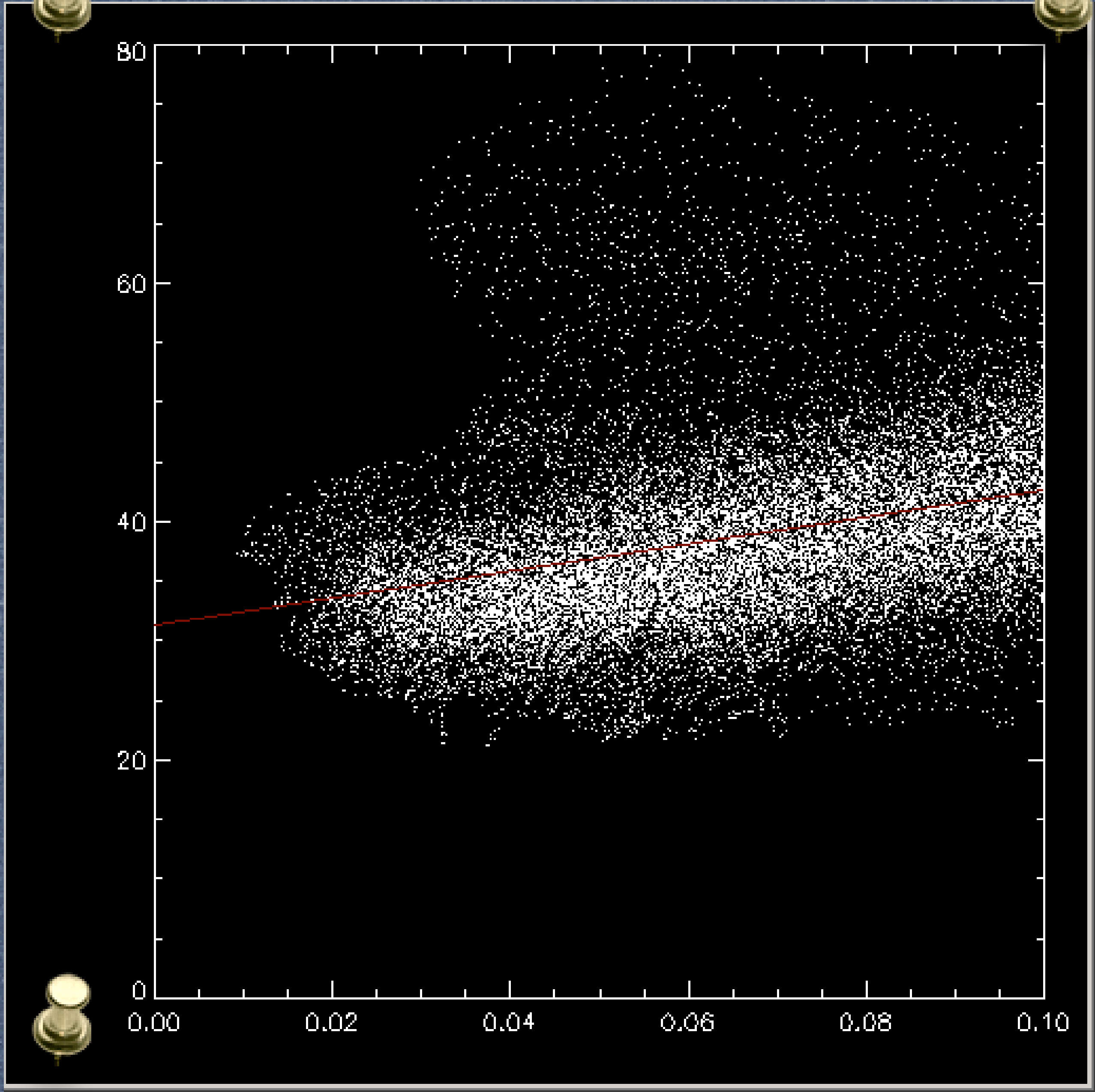
Using a brain segmentation, assuming a concentration ratio of 1/4 between WM and GM, we can define a 3D function Seg similar (in terms of frequency components) to the activity concentration C. Then we may roughly estimate PSF as:

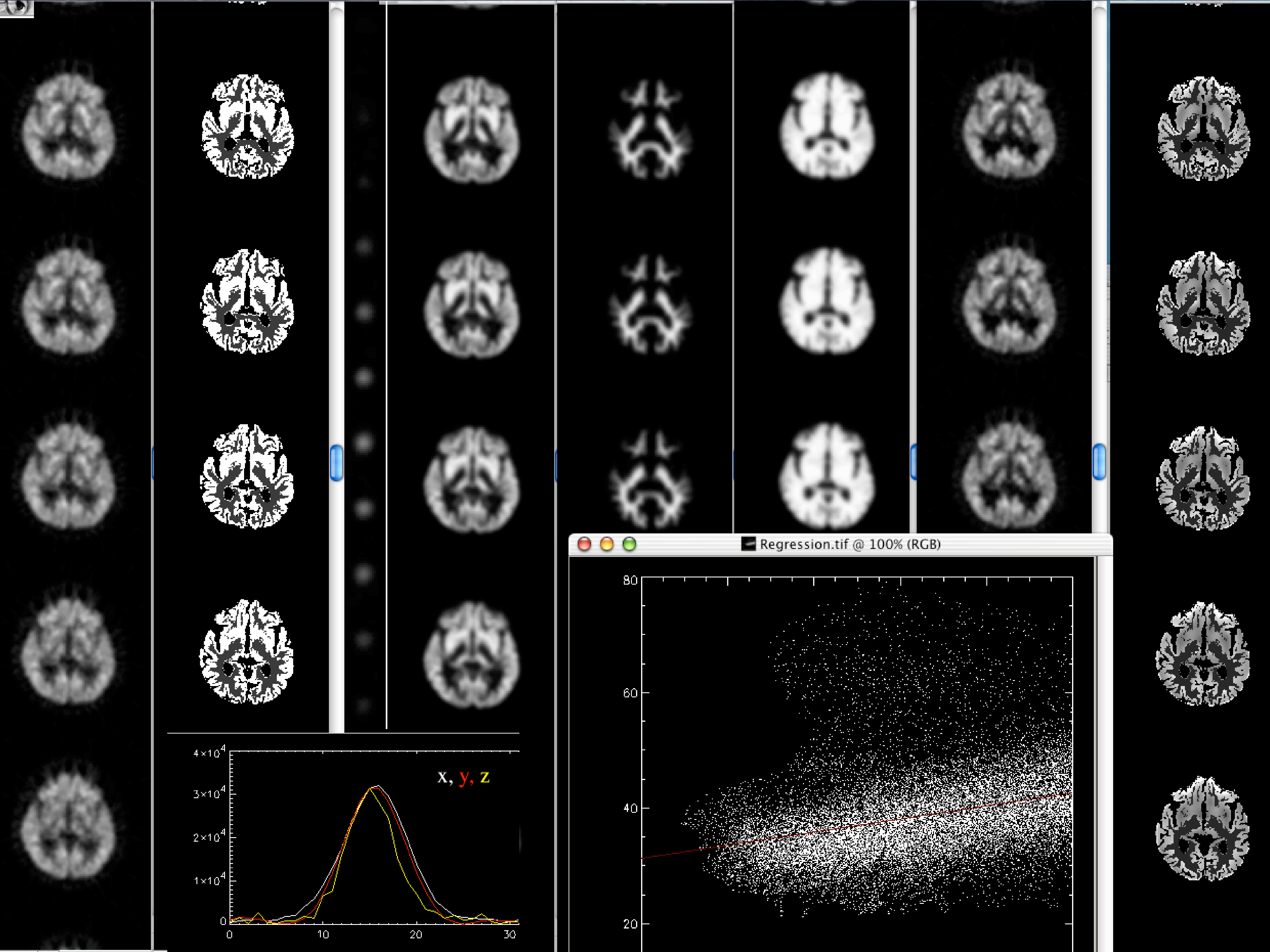
$$PSF \approx F^{-1}\left(\frac{\left|\hat{PET}\right|}{\left|\hat{Seg}\right|}\right)$$





3D PSF


80
60
40
20
0
0.00
0.02
0.04
0.06
0.08
0.10


x, y, z
4×10^4
3×10^4
2×10^4
1×10^4
0
0
10
20
30
Regression.tif @ 100% (RGB)
80
60
40
20