ChangDualSym2

Juillet 2017

Chang
Pour reconstruire la transformation de Radon Atténuée

$$(a, f) \to g = P_a(f) \tag{1}$$

$P_a(f)(s,\theta) = \int \exp(-D_a(x(t,s,\theta)))f(x(t,s,\theta)dt$

la méthode de Chang utilise l'inversion de la transformation de Radon Classique, en appliquant une correction au résultat selon

$$f(x) \simeq \frac{(P_0^{-1}g)(x)}{w_a(x)} \tag{2}$$

$w_a(x) = \frac{1}{2\pi}\int_0^{2\pi} \exp(-D_a(x,\theta))d\theta$

La méthode serait exacte si $\exp(-D_a(x,\theta))$ était indépendant de θ.

ChangDualSym2
La méthode "ChangDualSym2" est basée sur un principe analogue :

1. Si le poids $w(x,\theta)$ de la tranformation de Radon Pondérée

$$g = P_w(f)(s,\theta) = \int w(x(t,s,\theta),\theta)f(x(t,s,\theta))dt \tag{3}$$

était de la forme

$$w(x,\theta) = w_1(x)w_2(x.\theta^\perp,\theta) \tag{4}$$

on pourrait inverser sans erreur par

$$f(x) = \frac{1}{w_1(x)}P_0^{-1}(\frac{g(s,\theta)}{w_2(s,\theta)}) \tag{5}$$

2. La transformation de Radon Atténuée est une transformation de Radon Pondérée dont, en principe, le poids

$$w_a(x,\theta) = \exp(-D_a(x,\theta))$$

ne vérifie pas la propriété précédente.
On envisage de l'inverser de façon approximative en supposant que l'effet du poids $w_a(x,\theta)$ sur les projections n'est pas très différent de celui d'un poids $\tilde{w}(x,\theta)$ vérifiant la propriété

$$\tilde{w}(x,\theta) = w_1(x)w_2(x.\theta^\perp,\theta) \tag{6}$$

permettant d'obtenir une approximation de f par la formule

$$f(x) \simeq \tilde{f}(x) = \frac{1}{w_1(x)}P_0^{-1}(\frac{g(s,\theta)}{w_2(s,\theta)}) \tag{7}$$

Calcul de $\tilde{w}$

1. w_{sym}

$$w_{sym}(x,\theta) = \frac{1}{2}[w_a(x,\theta) + w_a(x,-\theta)]$$

est symétrique en θ. (cf article ....)

2. w_{sym_2}

$$w_{sym_2}(s,\theta) = moy\{w_{sym}(x(t,s,\theta)),t\}$$

moyenne sur la partie du segment $\{s\theta^\perp + t\theta, t\}$ contenue dans le support de a

3. w_{sym_3}

$$w_{sym_3}(x,\theta) = \frac{w_{sym}(x,\theta)}{w_{sym_2}(x.\theta^{\perp},\theta)}$$

4. w_{sym_1}

$$w_{sym_1}(x) = moy\{w_{sym_3}(x,\theta), \theta \in S^1\}$$

moyenne sur le cercle S^1.

5. $\tilde{w}$ (non calculé)

$$\tilde{w}(x,\theta) = w_{sym_1}(x) w_{sym_2}(x.\theta^{\perp},\theta)$$

Fantômes *phant*
On traite différents exemples avec Emission = ephant3.d
On compare les reconstructions par ChangDualSym2 à celles obtenues par Novmor et Chang0. Dans tous les cas il s'agit de **projections non bruitées**.

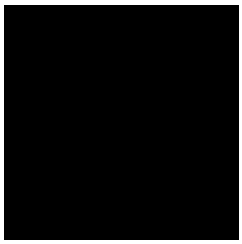
1. Att = aphantnoir.d	$a \equiv 0$	Proj = pphantnoir.d
2. Att = aphantblc.d	$a = 1$ dans $D(0,1)$	Proj = pphantblc.d
3. Att = aphantrad1.d	$a = 1$ dans $D(0,0.2)$ et $Cour(0,0.5,0.6)$	Proj = pphantrad1.d
4. Att = pphantrad10.d	$a = 10$ dans $D(0,0.2)$ et $Cour(0,0.5,0.6)$	Proj = pphantrad10.d
5. Att = aphantcstt.d	$a = dist(x,\{t\theta,t\})$ pour $\theta = \pi/4$	Proj = pphantcstt.d
6. Att = aphant3.d		Proj = pphant3.d

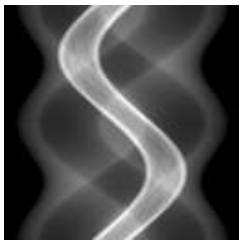
Reconstructions

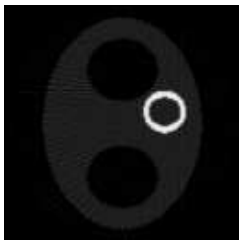
```
  aphantnoir.d    pphantnoir.d     cyclemor -i 6 E=0.21763
  aphantnoir.d    pphantnoir.d    cyclenchang -z E=0.19997
  aphantnoir.d    pphantnoir.d   cyclechdualsym2 E=0.19997

   aphantblc.d     pphantblc.d     cyclemor -i 6 E=0.22762
   aphantblc.d     pphantblc.d    cyclenchang -z E=0.19558
   aphantblc.d     pphantblc.d   cyclechdualsym2 E=0.20204

  aphantrad1.d    pphantrad1.d     cyclemor -i 6 E=0.21923
  aphantrad1.d    pphantrad1.d    cyclenchang -z E=0.24001
  aphantrad1.d    pphantrad1.d   cyclechdualsym2 E=0.20563

 aphantrad10.d   pphantrad10.d     cyclemor -i 6 E=0.25302
 aphantrad10.d   pphantrad10.d    cyclenchang -z E=0.49269
 aphantrad10.d   pphantrad10.d   cyclechdualsym2 E=0.41964

  aphantcstt.d    pphantcstt.d     cyclemor -i 6 E=0.22746
  aphantcstt.d    pphantcstt.d    cyclenchang -z E=0.27673
  aphantcstt.d    pphantcstt.d   cyclechdualsym2 E=0.20357

     aphant3.d       pphant3.d     cyclemor -i 6 E=0.23383
     aphant3.d       pphant3.d    cyclenchang -z E=0.2628
     aphant3.d       pphant3.d   cyclechdualsym2 E=0.24069
```

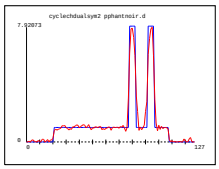
Images
Les reconstructions présentées ne concernent que celles obtenues avec ChangDualSym2.

1. pphantnoir.d

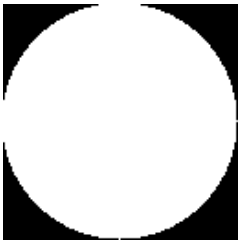

aphantnoir.d

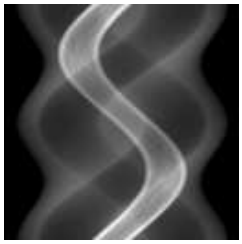

pphantnoir.d

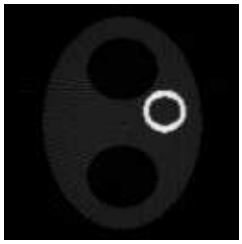

rphantnoir.d, E=0.199

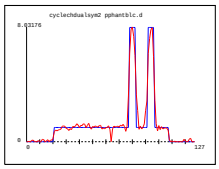

Coupe 64

2. pphantblc.d


aphantblc.d

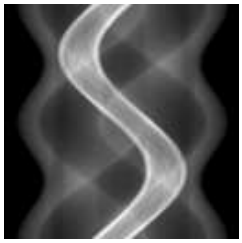

pphantblc.d

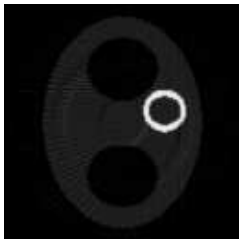

rphantblc.d, E=0.202

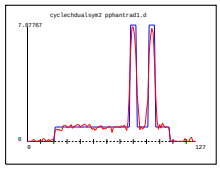

Coupe 64

3. pphantrad1.d


aphantrad1.d

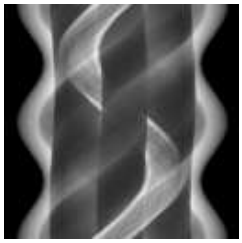

pphantrad1.d

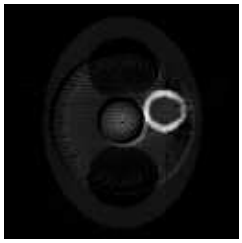

rphantrad1.d, E=0.205

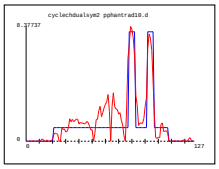

Coupe 64

4. pphantrad10.d

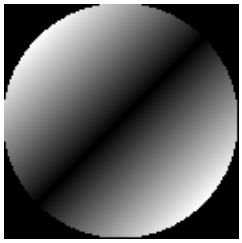

aphantrad10.d

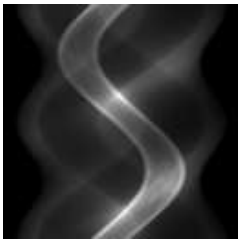

pphantrad10.d

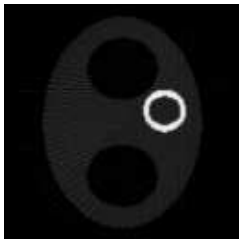

rphantrad10.d, E=0.419

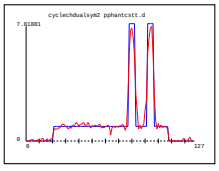

Coupe 64

5. pphantcstt.d


aphantcstt.d

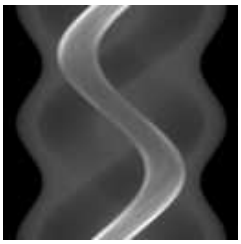

pphantcstt.d

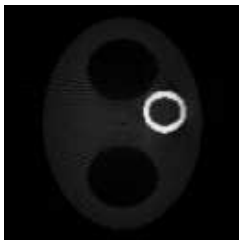

rphantcstt.d, E=0.203

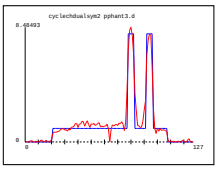

Coupe 64

6. pphant3.d


aphant3.d


pphant3.d


rphant3.d, E=0.240


Coupe 64

Commentaires
Les erreurs relatives et les coupes 64 des résultats sont assez satisfaisantes. On note néanmoins une anomalie au pixel (64,64)=(0,0) sur les coupes de rphantblc.d, rphantrad10.d et rphant3.d
En comparant rphantrad1.d et rphantrad10.d, on peut penser que cette anomalie, ne vient pas d'une erreur de programmation, mais plutôt d'un problème numérique lié aux valeurs de $a(x)$.
Fantômes *cylindre*
Comme ci-desus on traite différents exemples avec Emission = ecylindre2.d
On reprend les mêmes données d'atténuation, renommées *cylindre*, exception faite de la dernière qui est notre acylindre2.d habituel. On ajoute acylindre1t.d dont les valeurs sont le 1/3 de celles de acylindre2.d (Max=4 au lieu de Max=12).

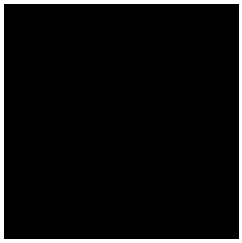
1. Att = acylindrenoir.d	$a \equiv 0$	Proj = pcylindrenoir.d
2. Att = acylindreblc.d	$a = 1$ dans $D(0,1)$	Proj = pcylindreblc.d
3. Att = acylindrerad1.d	$a = 1$ dans $D(0,0.2)$ et $Cour(0,0.5,0.6)$	Proj = pcylindrerad1.d
4. Att = pcylindrerad10.d	$a = 10$ dans $D(0,0.2)$ et $Cour(0,0.5,0.6)$	Proj = pcylindrerad10.d
5. Att = acylindrecstt.d	$a = dist(x,\{t\theta,t\})$ pour $\theta = \pi/4$	Proj = pcylindrecstt.d
6. Att = acylindre2.d		Proj = pcylindre2.d
7. Att = acylindre1t.d	$a = \frac{1}{3} acylindre2.d$	Proj = pcylindre1t.d

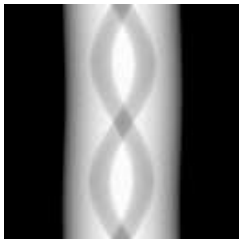
Reconstructions

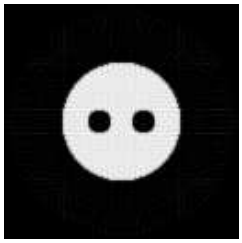
```
acylindrenoir.d pcylindrenoir.d     cyclemor -i 6 E=0.098347
acylindrenoir.d pcylindrenoir.d    cyclenchang -z E=0.095297
acylindrenoir.d pcylindrenoir.d   cyclechdualsym2 E=0.095297

 acylindreblc.d  pcylindreblc.d     cyclemor -i 6 E=0.10972
 acylindreblc.d  pcylindreblc.d    cyclenchang -z E=0.092968
 acylindreblc.d  pcylindreblc.d   cyclechdualsym2 E=0.099395

acylindrerad1.d pcylindrerad1.d     cyclemor -i 6 E=0.10188
acylindrerad1.d pcylindrerad1.d    cyclenchang -z E=0.15083
acylindrerad1.d pcylindrerad1.d   cyclechdualsym2 E=0.10298

acylindrerad10.d pcylindrerad10.d     cyclemor -i 6 E=0.1923
acylindrerad10.d pcylindrerad10.d    cyclenchang -z E=0.54272
acylindrerad10.d pcylindrerad10.d   cyclechdualsym2 E=0.4138

acylindrecstt.d pcylindrecstt.d     cyclemor -i 6 E=0.11117
acylindrecstt.d pcylindrecstt.d    cyclenchang -z E=0.20752
acylindrecstt.d pcylindrecstt.d   cyclechdualsym2 E=0.1006

   acylindre2.d    pcylindre2.d     cyclemor -i 6 E=0.10116
   acylindre2.d    pcylindre2.d    cyclenchang -z E=0.1443
   acylindre2.d    pcylindre2.d   cyclechdualsym2 E=0.22435

  acylindre1t.d   pcylindre1t.d     cyclemor -i 6 E=0.098411
  acylindre1t.d   pcylindre1t.d    cyclenchang -z E=0.11432
  acylindre1t.d   pcylindre1t.d   cyclechdualsym2 E=0.11428
```

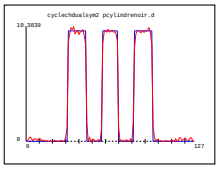
Images

1. pcylindrenoir.d

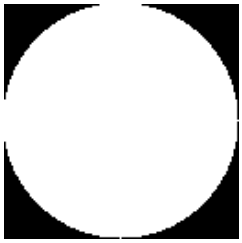

acylindrenoir.d

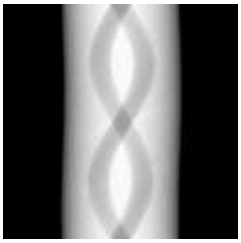

pcylindrenoir.d

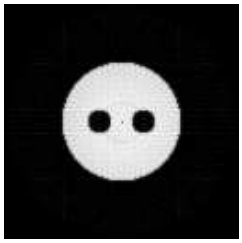

rcylindrenoir.d, E=0.095

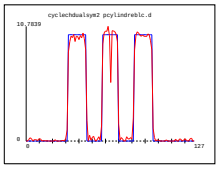

Coupe 64

2. pcylindreblc.d


acylindreblc.d

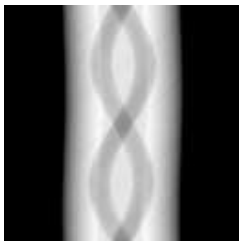

pcylindreblc.d

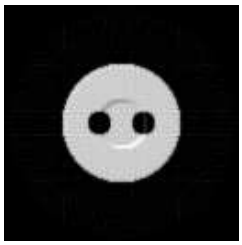

rcylindreblc.d, E=0.099

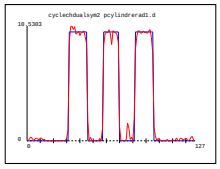

Coupe 64

3. pcylindrerad1.d


acylindrerad1.d

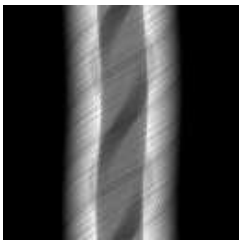

pcylindrerad1.d


rcylindrerad1.d, E=0.102

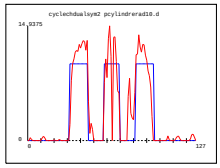

Coupe 64

4. pcylindrerad10.d

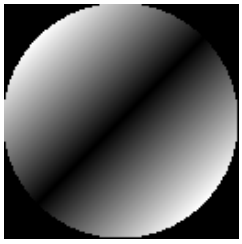

acylindrerad10.d

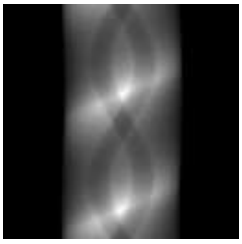

pcylindrerad10.d

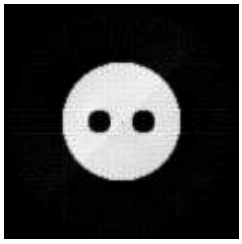

rcylindrerad10.d, E=0.413

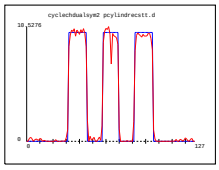

Coupe 64

5. pcylindrecstt.d

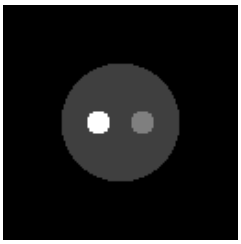

acylindrecstt.d

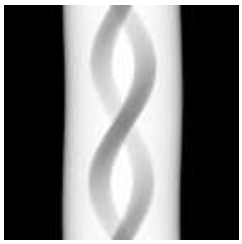

pcylindrecstt.d

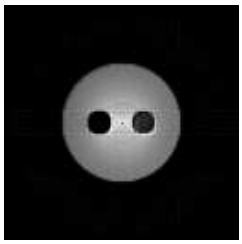

rcylindrecstt.d, E=0.100

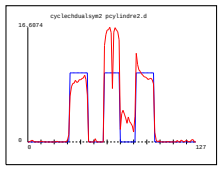

Coupe 64

6. pcylindre2.d


acylindre2.d

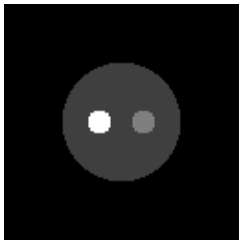

pcylindre2.d

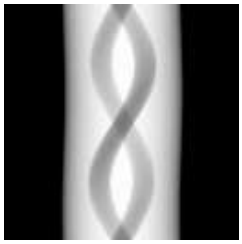

rcylindre2.d, E=0.224

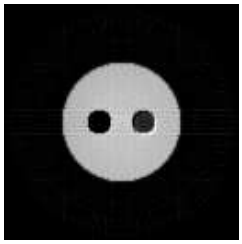


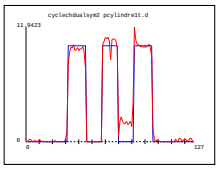
Coupe 64

7. pcylindre1t.d


acylindre1t.d


pcylindre1t.d


rcylindre1t.d, E=0.114



Coupe 64

Commentaires
Pour le cylindre les erreurs relatives et les coupes 64 des résultats sont assez satisfaisantes. Mais ici aussi on note une anomalie au pixel (64,64)=(0,0) sur les coupes de rcylindreblc.d, rcylindrerad10.d, rcylindrecstt.d et rcylindre2.d On constate sur rcylindre1t.d qu'en divisant les valeurs des pixels de acylindre2.d par 3 on obtient un résultat bien meilleur !