

Reconstruction d'image à partir de deux projections en tomographie X

Ali MOHAMMAD-DJAFARI

Laboratoire des Signaux et Systèmes (CNRS – SUPÉLEC – UPS).
 SUPÉLEC, Plateau de Moulon, 91192 Gif-sur-Yvette Cedex, France.
 E-mail: djafari@lss.supelec.fr

January 24, 2005

Abstract

Il s'agit d'un exemple simple pour montrer la nature mal-posé et les difficultés du l'inversion d'un problème de la reconstruction d'image à partir d'un nombre limité de projections en tomographie X.

1 Introduction

Considérons un problème de reconstruction d'image sous une forme discrétisée. Soient $\mathbf{f} = [f_1, \dots, f_{16}]^t$ les valeurs des pixels de l'image $f(x, y)$ de dimensions (4×4) et $\mathbf{p} = [p_1, \dots, p_8]^t$ les valeurs de ses projections $p(r, \phi)$ suivant les angles $\phi = 0$ et $\phi = 90$ degrés. Supposons $\Delta x = 1, \Delta y = 1, \Delta r = 1$ et considérons les deux représentations suivantes:

f_1	f_5	f_9	f_{13}	p_8	f_{11}	f_{12}	f_{13}	f_{14}	p_{24}
f_2	f_6	f_{10}	f_{14}	p_7	f_{21}	f_{22}	f_{23}	f_{24}	p_{23}
f_3	f_7	f_{11}	f_{15}	p_6	f_{31}	f_{32}	f_{33}	f_{34}	p_{22}
f_4	f_8	f_{12}	f_{16}	p_5	f_{41}	f_{42}	f_{43}	f_{44}	p_{21}
p_1	p_2	p_3	p_4		p_{11}	p_{12}	p_{13}	p_{14}	

Notons aussi

$$\mathbf{p}_1 = [p_1, \dots, p_4]^t = [p_{11}, \dots, p_{14}]^t,$$

$$\mathbf{p}_2 = [p_5, \dots, p_8]^t = [p_{21}, \dots, p_{24}]^t.$$

Formons les matrices $\mathbf{A}_1, \mathbf{A}_2$ et \mathbf{A} telle qu'on puisse écrire

$$\mathbf{p}_1 = \mathbf{A}_1 \mathbf{f}, \quad \mathbf{p}_2 = \mathbf{A}_2 \mathbf{f}, \quad \mathbf{p} = \mathbf{A} \mathbf{f} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} \mathbf{f}$$

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\mathbf{A}_2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Considérons l'image

$$\mathbf{f} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

et calculons sa projection:

```

A1=[
1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0;
0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0;
0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1];

A2=[
0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1;
0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0;
0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0;
1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0];

A=[A1;A2];
f=[0 0 0 0;0 1 1 0;0 1 1 0;0 0 0 0];
p=A*f(:);

```

$$\mathbf{p}^t = [0\ 2\ 2\ 0\ 2\ 2\ 0]$$

Ainsi, la résolution du problème direct ne pose aucune difficulté. Mais, considérons maintenant le problème inverse: étant donnée \mathbf{p} trouver \mathbf{f} .

Il est évident que ce problème a une infinité de solutions possibles. En voici quatre:

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} -.5 & 0 & 0 & .5 \\ 1 & 2 & 0 & -1 \\ -1 & 0 & 2 & 1 \\ 0.5 & 0 & 0 & -.5 \end{bmatrix} \quad \begin{bmatrix} -.5 & 0 & 0 & .5 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ .5 & 0 & 0 & -.5 \end{bmatrix}$$

Une première solution utilisée est la rétro-projection: $\hat{\mathbf{f}} = \mathbf{A}^t \mathbf{p}$. Exprimons alors la matrice \mathbf{A}^t :

$$\mathbf{A}^t = [\mathbf{A}_1^t \mid \mathbf{A}_2^t]$$

```
fh=A'*p; reshape(fh,4,4)
```

$$\hat{\mathbf{f}} = \begin{bmatrix} 0 & 2 & 2 & 0 \\ 2 & 4 & 4 & 2 \\ 2 & 4 & 4 & 2 \\ 0 & 2 & 2 & 0 \end{bmatrix}$$

Notons que $\hat{\mathbf{f}} = \mathbf{A}^t \mathbf{p} = \mathbf{A}_1^t \mathbf{p}_1 + \mathbf{A}_2^t \mathbf{p}_2$ est l'addition de deux images

$$\hat{\mathbf{f}} = \begin{bmatrix} 0 & 2 & 2 & 0 \\ 0 & 2 & 2 & 0 \\ 0 & 2 & 2 & 0 \\ 0 & 2 & 2 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

chacune étant la rétro-projection d'une des deux projections.

Remarquons aussi que cette image, à une constant près est très proche du résultat de la convolution de l'image d'origine avec la réponse impulsionnelle

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

En vue de la définition d'une solution au sens d'inversion généralisée, calculons les matrices $\mathbf{A}^t \mathbf{A}$ et $\mathbf{A} \mathbf{A}^t$:

$$\mathbf{A}^t = [\mathbf{A}_1^t \mid \mathbf{A}_2^t]$$

$$\mathbf{A} \mathbf{A}^t = \left[\begin{array}{c|c} \mathbf{A}_1 \mathbf{A}_1^t & \mathbf{A}_1 \mathbf{A}_2^t \\ \hline \mathbf{A}_2 \mathbf{A}_1^t & \mathbf{A}_2 \mathbf{A}_2^t \end{array} \right]$$

$$\mathbf{A} \mathbf{A}^t = \left[\begin{array}{c|c} 4\mathbf{I} & \mathbf{1} \\ \hline \mathbf{1} & 4\mathbf{I} \end{array} \right] = \begin{bmatrix} 4 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 4 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 4 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 4 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 4 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 4 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 4 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 4 \end{bmatrix}$$

$$\mathbf{A}^t \mathbf{A} = \left[\begin{array}{c|c} \mathbf{A}_1^t \mathbf{A}_1 & \mathbf{A}_2^t \mathbf{A}_1 \\ \hline \mathbf{A}_1^t \mathbf{A}_2 & \mathbf{A}_2^t \mathbf{A}_2 \end{array} \right]$$

$$\mathbf{A}_1^t \mathbf{A}_1 = \mathbf{A}_2^t \mathbf{A}_2 = \left[\begin{array}{c|c} \mathbf{1} + \mathbf{I} & \mathbf{I} \\ \hline \mathbf{I} & \mathbf{1} + \mathbf{I} \end{array} \right]$$

$$\mathbf{A}_1^t \mathbf{A}_2 = \mathbf{A}_2^t \mathbf{A}_1 = \left[\begin{array}{c|c} \mathbf{I} & \mathbf{I} \\ \hline \mathbf{I} & \mathbf{I} \end{array} \right],$$

$$\mathbf{A}^t \mathbf{A} = \begin{bmatrix} 2 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 2 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 2 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 2 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 2 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 2 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 2 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 2 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 2 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 \end{bmatrix}$$

Calculons les valeurs singulières des matrices $\mathbf{A} \mathbf{A}^t$ et $\mathbf{A}^t \mathbf{A}$:

$$\mathbf{A} \mathbf{A} \mathbf{A}^t = \mathbf{A}^t \mathbf{A} \mathbf{A}^t; \text{svd}(\mathbf{A} \mathbf{A} \mathbf{A}^t)$$

$$\text{svd}(\mathbf{A} \mathbf{A} \mathbf{A}^t) = [8 \ 4 \ 4 \ 4 \ 4 \ 4 \ 4 \ 4 \ 0]$$

$$\mathbf{A}^t \mathbf{A} \mathbf{A} = \mathbf{A} \mathbf{A}^t \mathbf{A}; \text{svd}(\mathbf{A}^t \mathbf{A} \mathbf{A})$$

$$\text{svd}(\mathbf{A}^t \mathbf{A} \mathbf{A}) = [8 \ 4 \ 4 \ 4 \ 4 \ 4 \ 4 \ 4 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$$

Notons que ces deux matrices sont singulières. Rappelons qu'une solution au sens des moindres carrés s'écrit:

$$\hat{\mathbf{f}} = \arg \min \mathbf{f} \| \mathbf{p} - \mathbf{A} \mathbf{f} \|^2 = (\mathbf{A}^t \mathbf{A})^{-1} \mathbf{A}^t \mathbf{p}$$

si la matrice $\mathbf{A}^t \mathbf{A}$ était inversible.

De même, une solution de norme minimale est

$$\hat{\mathbf{f}} = \arg \min \mathbf{A} \mathbf{f} = \mathbf{p} \| \mathbf{f} \|^2 \longrightarrow \hat{\mathbf{f}} = \mathbf{A}^t (\mathbf{A} \mathbf{A}^t)^{-1} \mathbf{p}$$

Nous ne pouvons alors calculer aucune de ces deux solutions, mais notons cependant:
`fh=diag(1./diag(AtA))*A'*p; reshape(fh,4,4)`

$$\hat{\mathbf{f}} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 2 & 2 & 1 \\ 1 & 2 & 2 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

`fh=A'*diag(1./diag(AAt))*p; reshape(fh,4,4)`

$$\hat{\mathbf{f}} = \begin{bmatrix} 0 & .5 & .5 & 0 \\ .5 & 1 & 1 & .5 \\ .5 & 1 & 1 & .5 \\ 0 & .5 & .5 & 0 \end{bmatrix}$$

Il est cependant possible de calculer la solution inverse généralisée qui est la solution de norme minimale de $\mathbf{A}^t \mathbf{A} \mathbf{f} = \mathbf{A}^t \mathbf{p}$ en utilisant la décomposition tronquée des valeurs singulières:

$$\hat{\mathbf{f}} = \sum_{k=1}^K \frac{\langle \mathbf{p}, \mathbf{u}_k \rangle}{\lambda_k} \mathbf{v}_k$$

où \mathbf{u}_k et \mathbf{v}_k sont, respectivement, des vecteurs propres de $\mathbf{A} \mathbf{A}^t$ et de $\mathbf{A}^t \mathbf{A}$ et λ_k sont des valeurs singulières associées.

```
[U,S,V]=svd(A);
s=diag(S);s1=[1./s(1:7);zeros(1,1)];
S1=[diag(s1);zeros(8,8)];
fh=V*S1*U'*p;reshape(fh,4,4)
```

Dans cet exemple $K = 7$. Notons aussi que le noyau de la transformation linéaire $\mathbf{p} = \mathbf{A} \mathbf{f}$, *i.e.*, $\{\mathbf{f} | \mathbf{A} \mathbf{f} = 0\}$ est

$$\mathbf{V}(\mathbf{I} - \mathbf{S}^+ \mathbf{S})\mathbf{z} = \sum_{k=K+1}^N z_k \mathbf{v}_k$$

avec \mathbf{z} un vecteur arbitraire.

```
fh=svdpca(A,p,.1,7);reshape(fh,4,4)
```

$$\hat{\mathbf{f}} = \begin{bmatrix} -0.2500 & 0.2500 & 0.2500 & -0.2500 \\ 0.2500 & 0.7500 & 0.7500 & 0.2500 \\ 0.2500 & 0.7500 & 0.7500 & 0.2500 \\ -0.2500 & 0.2500 & 0.2500 & -0.2500 \end{bmatrix}$$

ou encore par l'algorithme itératif suivant:

```
for k=1:100;
fh=fh+.1*A'*(p-A*fh(:));
end;
reshape(fh,4,4)
```

$$\hat{\mathbf{f}} = \begin{bmatrix} -0.2500 & 0.2500 & 0.2500 & -0.2500 \\ 0.2500 & 0.7500 & 0.7500 & 0.2500 \\ 0.2500 & 0.7500 & 0.7500 & 0.2500 \\ -0.2500 & 0.2500 & 0.2500 & -0.2500 \end{bmatrix}$$

Notons que, $\mathbf{A}^t \mathbf{A} + \lambda \mathbf{I}$ et $\mathbf{A} \mathbf{A}^t + \lambda \mathbf{I}$ sont inversibles pour $\lambda > 0$. Ceci nous permet de calculer

```
lambda=.01;
fh=inv(AtA+lambda*eye(size(AtA)))*(A'*p);
reshape(fh,4,4)
```

$$\hat{\mathbf{f}} = \begin{bmatrix} -0.2491 & 0.2497 & 0.2497 & -0.2491 \\ 0.2497 & 0.7484 & 0.7484 & 0.2497 \\ 0.2497 & 0.7484 & 0.7484 & 0.2497 \\ -0.2491 & 0.2497 & 0.2497 & -0.2491 \end{bmatrix}$$

ou encore

```
lambda=.01;
fh=A'*inv(AAt+lambda*eye(size(AAt)))*p;
reshape(fh,4,4)
```

qui fourni la même solution.

On peut remarquer qu'il n'y aura aucune méthode linéaire qui puisse fournir la solution exacte. Par contre, l'emploi d'une méthode non linéaire peut permettre d'obtenir cette solution. Par exemple imposer à la solution d'être positive peut permettre de trouver une solution qui sera plus proche de la vraie solution.

```
for k=1:100
fh=fh+.1*A'*(p-A*fh(:));
fh=fh.*(fh>0);
end
reshape(fh,4,4);
```

$$fh = \begin{bmatrix} 0 & 0.0000 & 0.0000 & 0 \\ 0.0000 & 1.0000 & 1.0000 & 0.0000 \\ 0.0000 & 1.0000 & 1.0000 & 0.0000 \\ 0 & 0.0000 & 0.0000 & 0 \end{bmatrix}$$

Examinons maintenant le cas d'une image de plus grande taille (256×256). Ici, il n'est pas question de former la matrice \mathbf{A} car de dimensions ($256^2 \times 256$). Par contre, on aura besoin de calculer $\mathbf{A}\mathbf{f}$ et $\mathbf{A}^t\mathbf{p}$, mais pour cela on n'a pas besoin de construire réellement la matrice \mathbf{A} . Les deux fonctions suivantes effectuent ces deux tâches :

```
function p=direct(f);
p1=sum(f);
p2=sum(f');
p=[p1(:);p2(:)];
return

function f=transp(p);
l=length(p);p1=p(1:l/2);p2=p(l/2+1:l);
f=ones(l/2,1)*p1'+p2*ones(1,l/2);
return
```

Nous pouvons alors utiliser les méthodes itératives :

Moindre carré avec contrainte de positivité :

```
alpha=.1;
for k=1:100
g=trans(p-direct(fh);
fh=fh+alpha*g;
fh=fh.*(fh>0);
end
```

Régularisation quadratique avec contrainte de positivité :

```
alpha=.1;d=[-1 0 -1;0 4 0;-1 0 -1];
for k=1:100
g0=trans(p-direct(fh);
g=g0-lambda*conv2(fh,d,'same');
fh=fh+alpha*g;
fh=fh.*(fh>0);
end
```

Remarquons que les algorithmes présentés plus haut sont assez rudimentaires (gradient à pas constant et à nombre d'itérations fini). Il est évident que l'on peut faire mieux. A titre d'exemple, nous avons développé un logiciel d'optimisation (gpave) un peu plus élaboré qui met en œuvre d'autres algorithmes d'optimisation, comme par exemple, gradient à pas adaptative, gradient conjugué et d'autres encore. Les lignes de codes Matlab qui suivent montrent l'usage de ce logiciel. Il faut tout d'abord écrire deux routines qui calculent le critère `crit` qui calcule $J = \|\mathbf{p} - \mathbf{A}\mathbf{f}\|^2 + \lambda\|\mathbf{D}\mathbf{f}\|^2$ et son gradient `dcrit` $\nabla J = -2\mathbf{A}^t(\mathbf{p} - \mathbf{A}\mathbf{f}) + 2\lambda\mathbf{D}^t\mathbf{D}\mathbf{f}$ où $\mathbf{D}\mathbf{f}$ correspond à l'application d'une opération de convolution de l'image $f(i, j)$ avec une réponse impulsionnelle $\begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}$ ce qui correspond à

$$\sum_i \sum_j (|f(i, j) - f(i-1, j)|^2 + |f(i, j) - f(i, j-1)|^2).$$

Notez aussi que $\mathbf{D}^t\mathbf{D}\mathbf{f}$ correspond à l'application d'une opération de convolution de l'image $f(i, j)$ avec une réponse impulsionnelle

$$\begin{bmatrix} -1 & 0 & -1 \\ 0 & -4 & 0 \\ -1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} * \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}$$

où $*$ signifie une convolution.

```
function J=crit(fh,p,lambda)
dp=p-direct(fh);
J0=sum(dp(:).^2);
d=[-1 1;1 -1];
df=conv2(fh,d,'same');
J1=sum(df(:).^2);
J=J0+lambda*J1;
return
```

```
function dJ=dcrit(fh,p,lambda)
dp=p-direct(fh);
dJ0=-2*transp(dp);
d=[-1 0 -1;0 4 0;-1 0 -1];
dJ1=conv2(fh,d,'same');
dJ=dJ0+lambda*dJ1;
return
```

```

f0=transp(p);
options = goptions;
lambda=1;
fh=gpav('crit',f0,options,'dcrit',p,lambda);
reshape(fh,4,4)

```

Avec ce programme d'optimisation il est alors facile de modifier les routines `crit` et `dcrit` pour changer le critère de la régularisation. Les figures suivantes montrent un certain nombre des résultats.

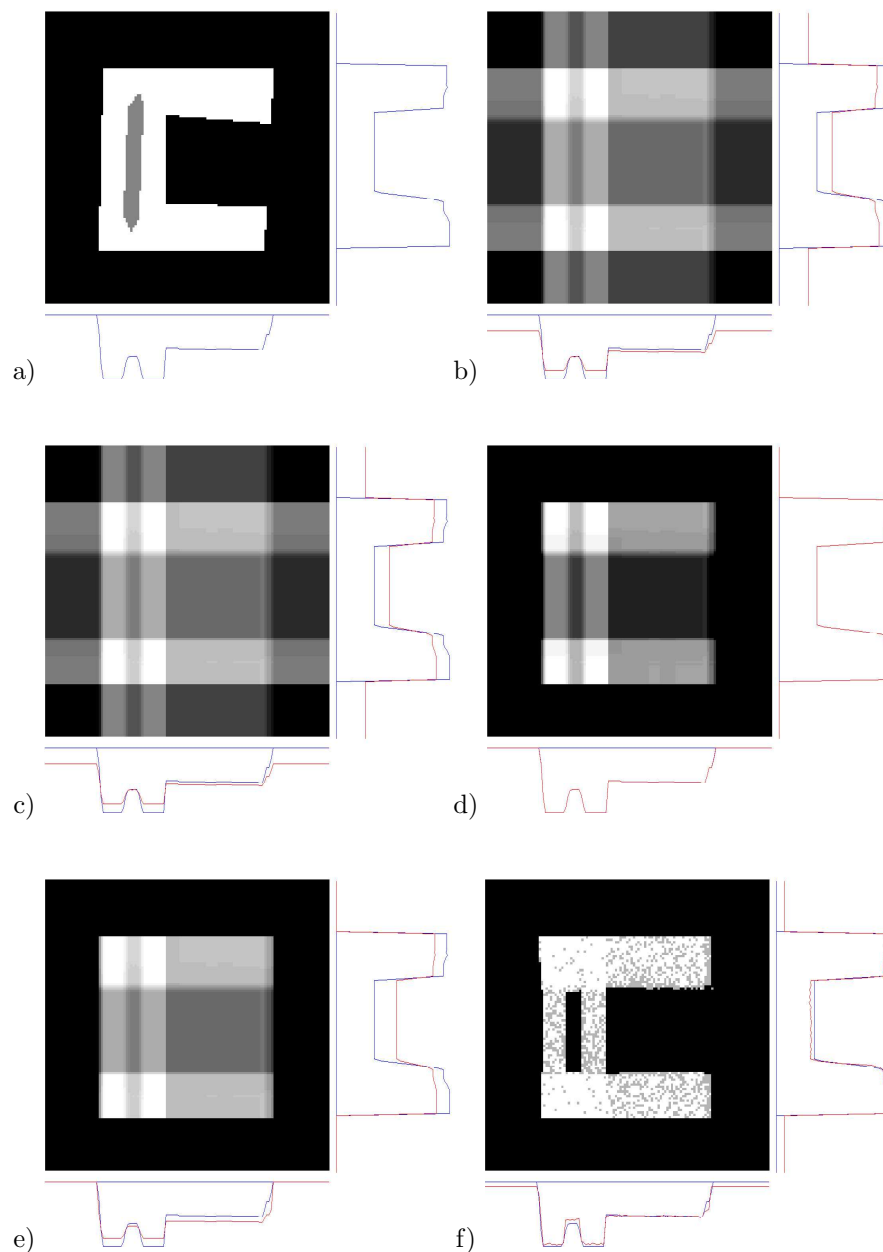


Figure 1: Reconstruction à partir de 2 projections: a) l'objet original et les données, b) rétroprojection c) rétroprojection filtré avec contrainte d) MC avec positivité, e) Régularisation quadratique avec positivité f) Résultat obtenu avec la méthode `multi-level-set`.

On remarque que le problème est très mal-conditionnée au sens que la manque d'information dans les données est trop important. Il faut pouvoir obtenir d'autres données, *i.e.*, des projections suivant d'autres angles. Les figures suivantes montrent des exemples de reconstructions pour le cas où on a 5 projections.