Filtered BackProjection in parallel beam geometry as the back projection of the derivation of parallel Hibert transform of the projections

L. Desbat - Laurent.Desbat@imag.fr

M2 MIA - GICAO - UFR IMA

1 Introduction

The aim of this “TP” (“Practical Work”) is to rewrite the FilterBackprojection as the BackProjection of the derivation of parallel Hibert transform of the projections. This means that we want to replace the “ramp” filtering of the projections by the Hilbert filtering of the projections followed by the derivation of the projections. The algorthim is then ended by the same backprojection algorithm as usually. More precisely, let μ be the attenuation function to be reconstructed by its parallel projections :

$$p(\phi, s) \overset{\text{def}}{=} \mathcal{X}\mu\left(\vec{\zeta}, s\vec{\theta}\right) = \mathcal{R}\mu\left(\vec{\theta}, s\right) = \int_{\mathbb{R}} \mu\left(l\vec{\zeta} + s\vec{\theta}\right) dl, \tag{1}$$

where $\vec{\theta} = (\cos\phi, \sin\phi)\,, \vec{\zeta} = (-\sin\phi, \cos\phi)\,, \phi \in [0, 2\pi)$. We remark that when $n = 2$, $\mathcal{X}\mu\left(\vec{\zeta}, s\vec{\theta}\right) = \mathcal{R}\mu\left(\vec{\theta}, s\right)$.

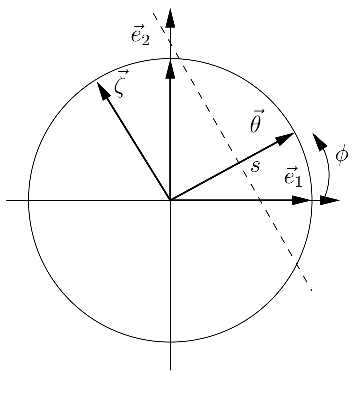

FIGURE 1 – 2D tomography : parallel geometry parameters. The line of integration $s\vec{\theta} + \mathbb{R}\vec{\zeta}$ is the dashed line.

The FBP with the ramp filter can be decomposed in two step

1. The **filtering step** by the ramp filter

$$\forall\phi, g_F(\phi,s) = \int_{\mathbb{R}} \hat{p}_\phi(\sigma)|\sigma|e^{2i\pi\sigma s}d\sigma$$

   Generally the filtering in the Fourier space $|\sigma|$ called the ramp filtering (denoted by r here) is replaced by more regular (and less sensitive to noise) filters such as $\hat{r}_c(\sigma) = \chi_{[-c,c]}(\sigma)|\sigma|$, where $c > 0$ is a cut-off frequency, i.e,.

$$r_c(s) = \int_{-c}^{c} \chi_{[-c,c]}(\sigma)|\sigma|e^{2i\pi\sigma s}d\sigma$$

   Thus p_F is the convolution of p_ϕ by r, i.e. $p_F(\phi,s) = r_c \star p_\phi(s)$, more precisely

$$p_F(\phi,s) = \int_{\mathbb{R}} r_c(s-u)p_\phi(u)du.$$

   Note thas r_c is a non-local filter with a long dependancy.

2. The filtering step is followed by the **back projection** step

$$\mu(\vec{x}) = \int_0^\pi p_F(\phi, \vec{x}\cdot\theta)d\phi.$$

We have remarked that the ramp filter response $|\sigma| = \frac{1}{2\pi}(2i\pi\sigma)\left(-i\mathrm{sgn}(\sigma)\right)$. Thus the ramp filter is the Hilbert filtering (multiplication of by $\hat{p}_\phi(\sigma)$ by $-i\mathrm{sgn}(\sigma)$) composed by the derivation (multiplication by $2i\pi\sigma$). The filtering step can be thus written $p_H(\phi,s) = Hp_\phi(s)$ see (2) followed by a derivation $p_F(\phi,s) = \frac{\partial p_H}{\partial s}(\phi,s)$. Here also a regularized Hilbert filtering is used in practice, see section A.1.

2 Work to do

The aim of the TP is to replace the filtering step (filter r_c) by the Hilbert filtering of the projections $p_\phi(s)$ (for all ϕ, ϕ being fixed) followed by a derivation according to s (for all ϕ, ϕ being fixed).

We define p_H the Hilbert transform of the parallel projection p

$$p_H(\phi,s) = \int_{-\infty}^{+\infty} p(\phi,u)h(s-u)du \text{ where } h(u) = \frac{1}{\pi u} \tag{2}$$

and $\hat{h}(\sigma) = -i\mathrm{sgn}(\sigma)$ (distribution). The ramp filtering p_R of p is

$$p_R(\phi,s) = \frac{1}{2\pi}\frac{\partial}{\partial s}p_H(\phi,s)$$

Thus in the given matlab code `BackDerivHilbertPARstudent.m` based on a classical FBP code, we have just to replace the ramp filtering step by computing p_H according to eq. (2) followed by the derivation of the p_H according to the variable s.

Hints :

1. *all the work to be done is between the lines*
   `% BEGIN FILTERING`
   and
   `% END FILTERING`

2. *the matlab code for computing the Hilbert filter with cutoff frequency* $c > 0$ *is given, see the file* `hilbertfilter.m`. `Hfilter = hilbert-filter(ndetecteurs,hs,freqcutoff)` *returns a vector* `Hfilter` *of length* `2*ndetecteurs-1`. *In order to filter the projections p by* `Hfil-ter` *you can "zeropadd" each projection p_ϕ for each ϕ, then make a circular convolution (of length $2n_s$) of the "zeropadded" projection p_ϕ by the* `Hfilter` *(see the fast matlab command* `cconv`*) and take the result by unzeropadding. In the following, for all* `k`
   ```
   zeropaddedproj=[sino(k,:) zeros(1,ns-1)]; % zeropadding
   paddedHproj=cconv(zeropaddedproj, hfilter, 2*ns-1); % cir-
   cular convolution
   Hsino(k,:)=paddedHproj(1:ns); % unzeropadding
   ```

A Numerics

A.1 The Hibert Filter

For numerical applications, we consider a frequency cut-off version of the Hibert filtering (Low Pass approximation of the Hiber filter). We define h_{Hc} such that $\hat{h}_{Hc}(\sigma) = \chi_{[-c,c]}(\sigma)(-i\text{sgn}(\sigma))$ where $c > 0$ is a cut-off frequency parameter. Then

$$h_{Hc}(s) = \int_{-c}^{c} -i\text{sgn}(\sigma)\mathrm{e}^{2\mathrm{i}\pi\sigma\mathrm{s}}\mathrm{d}\sigma = \frac{1-\cos(2\pi\mathrm{cs})}{\pi\mathrm{s}}.$$

A.2 The derivation

Let p be a 2D projection, i.e. $p(\phi, s)$ be a sinogram, regularly (in s at least) sampled such that $\mathbf{p}(k,l) = p(\phi_k, s_l)$then the partial derivative in can be approximated by

$$\frac{\partial}{\partial s} p\left(\phi_k, s_j\right) \approx \frac{p\left(\phi_k, s_{j+1}\right) - p\left(\phi_k, s_{j-1}\right) -}{2h_s}$$

or by

$$\frac{\partial}{\partial s} p\left(\phi_k, s_j\right) \approx \frac{p\left(\phi_k, s_{j+1}\right) - p\left(\phi_k, s_j\right) -}{h_s}$$

where h_s is the sampling step in s $(h_s = s_{j+1} - s_j)$.