Part III

Appendices

Appendix A

The Dirac Delta Function

The Dirac delta function $\delta(\cdot)$ was introduced by Dirac [83]. The function is a part of a new class of functions known as generalized functions. A good reference for generalized functions is [140].

The delta function is special in the sense that it is defined from

$$\delta(x) = 0 \quad \text{for } x \neq 0 \tag{A.1}$$

$$\int_{-\infty}^{\infty} \delta(x)\ dx = 1 \tag{A.2}$$

The most important property of the delta function comes is that

$$\int_{-\infty}^{\infty} g(x)\ \delta(a\,x+b)\ dx = \frac{1}{|a|}\int_{-\infty}^{\infty} g\left(\frac{x-b}{a}\right)\ \delta(x)\ dx = \frac{1}{|a|} g\left(\frac{-b}{a}\right) \tag{A.3}$$

which also shows that the delta function is an even function of its argument $\delta(x) = \delta(-x)$.

If the delta function has a general function as argument is can be rewritten [73]

$$\delta(f(x)) = \sum_{i=1}^{I} \frac{\delta(x-x_i)}{|f'(x_i)|} \tag{A.4}$$

where x_i is the zeros of the function $f(x)$. Thus it is found that

$$\int_{-\infty}^{\infty} g(x)\delta(f(x))\ dx = \sum_{i=1}^{I} \frac{g(x_i)}{|f'(x_i)|} \tag{A.5}$$

This result is very important with respect to the Radon transform and especially to the generalized Radon transform, where the argument of the delta function determines the curve shape.

The delta function is not a normal function and often approximations that converge to the delta function are quite useful, e.g.

$$\psi(x) = \frac{1}{\sqrt{\pi}a} e^{-\left(\frac{x}{a}\right)^2} \tag{A.6}$$

and in the limit

$$\lim_{a\to 0} \psi(x) = \delta(x) \tag{A.7}$$

The function $\psi(\cdot)$ can be very valuable, because it easily can be analyzed with conventional calculus and its derivatives can also be used to approximate the derivatives of the delta function.

$$\lim_{a\to 0} \psi^{(n)}(x) = \delta^{(n)}(x) \tag{A.8}$$

Appendix B

Properties of the Normal Radon Transformation

B.1 Basic Properties of the Radon Transform

In this section several basic properties of the normal Radon transform are shown. All of the following based on the properties of the δ-function reviewed in Appendix A.

$$\check{g}(\rho,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)\delta(\rho - x\cos\theta - y\sin\theta)\ dx\ dy \tag{B.1}$$

B.1.1 Linearity

The first property is that the normal Radon transform is linearity. If α_q is an array of constants, then

$$g(x,y) = \sum_q \alpha_q\ g_q(x,y) \Rightarrow \check{g}(\rho,\theta) = \sum_q \alpha_q\ \check{g}_q(\rho,\theta) \tag{B.2}$$

B.1.2 Shifting

Assume that a function $g(x,y)$ is shifted

$$h(x,y) = g(x-x_0, y-y_0) \ \Rightarrow \tag{B.3}$$

$$\check{h}(\rho,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x-x_0,y-y_0)\ \delta(\rho - x\cos\theta - y\sin\theta)\ dx\ dy \tag{B.4}$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(\tilde{x},\tilde{y})\ \delta((\rho - x_0\cos\theta - y_0\sin\theta) - \tilde{x}\cos\theta - \tilde{y}\sin\theta)\ d\tilde{x}\ d\tilde{y} \tag{B.5}$$

$$= \check{g}(\rho - x_0\cos\theta - y_0\sin\theta, \theta) \tag{B.6}$$

Note that only the ρ-coordinate is changed.

B.1.3 Rotation

Here $g(x,y)$ is expressed in polar form, i.e., $g(x,y) = g(r,\phi)$. In this case rotation is fairly easy

$$h(r,\phi) = g(r,\phi-\phi_0) \tag{B.7}$$

$$\check{h}(\rho,\theta) = \int_{-\infty}^{\infty}\int_{0}^{\pi} g(r,\phi-\phi_0)\ \delta(\rho - r\cos\phi\cos\theta - r\sin\phi\sin\theta)\ |r|\ d\phi\ dr \tag{B.8}$$

$$= \int_{-\infty}^{\infty}\int_{0}^{\pi} \check{g}(r,\tilde{\phi})\ \delta(\rho - r\cos(\theta - \tilde{\phi} - \phi_0))\ |r|\ d\tilde{\phi}\ dr \tag{B.9}$$

$$= \check{g}(\rho,\theta-\phi_0) \tag{B.10}$$

This is quite obvious. If the coordinate system (x,y) is turned ϕ_0, then the Radon transform is also turned ϕ_0.

B.1.4 Scaling

Assume a scaling in both coordinates

$$h(x,y) = g\left(\frac{x}{a}, \frac{y}{b}\right), \; a > 0 \text{ and } b > 0 \tag{B.11}$$

In this case the Radon transform can be rewritten

$$\begin{aligned}
\check{h}(\rho,\theta) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g\left(\frac{x}{a}, \frac{y}{b}\right) \; \delta(\rho - x\cos\theta - y\sin\theta) \; dx \; dy && \text{(B.12)}\\
&= ab \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(\tilde{x},\tilde{y}) \; \delta(\rho - a\tilde{x}\cos\theta - b\tilde{y}\sin\theta) \; d\tilde{x} \; d\tilde{y} && \text{(B.13)}\\
&= \frac{ab}{|\gamma|} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(\tilde{x},\tilde{y}) \; \delta\left(\frac{\rho}{\gamma} - \tilde{x}\frac{a\cos\theta}{\gamma} - \tilde{y}\frac{b\sin\theta}{\gamma}\right) \; d\tilde{x} \; d\tilde{y} && \text{(B.14)}\\
&\equiv \frac{ab}{|\gamma|} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(\tilde{x},\tilde{y}) \; \delta(\tilde{\rho} - \tilde{x}\cos\tilde{\theta} - \tilde{y}\sin\tilde{\theta}) \; d\tilde{x} \; d\tilde{y} && \text{(B.15)}\\
&= \frac{ab}{|\gamma|} \; \check{g}(\tilde{\rho},\tilde{\theta}) && \text{(B.16)}
\end{aligned}$$

Now $\tilde{\rho}$ and $\tilde{\theta}$ must be found as functions of ρ and θ. From the equations above it is found that

$$\begin{aligned}
\cos\tilde{\theta} &= \frac{a\cos\theta}{\gamma} \text{ and } \sin\tilde{\theta} = \frac{b\sin\theta}{\gamma} && \text{(B.17)}\\
\cos^2\tilde{\theta} + \sin^2\tilde{\theta} &= 1 \;\Rightarrow\; \gamma = \sqrt{(a\cos\theta)^2 + (b\sin\theta)^2} \;\Rightarrow && \text{(B.18)}\\
\tilde{\rho} &= \frac{\rho}{\gamma} = \frac{\rho}{\sqrt{(a\cos\theta)^2 + (b\sin\theta)^2}} && \text{(B.19)}\\
a\tan\tilde{\theta} &= b\tan\theta \;\Rightarrow\; \tilde{\theta} = \arctan\left(\frac{b}{a}\tan\theta\right) && \text{(B.20)}
\end{aligned}$$

To summarize the equations shown above

$$\begin{aligned}
h(x,y) &= g\left(\frac{x}{a}, \frac{y}{b}\right), \; a > 0 \text{ and } b > 0 \;\Rightarrow && \text{(B.21)}\\
\check{h}(\rho,\theta) &= \frac{ab}{\gamma} \; \check{g}\left(\frac{\rho}{\gamma}, \arctan\left(\frac{b}{a}\tan\theta\right)\right) && \text{(B.22)}\\
\gamma &= \sqrt{(a\cos\theta)^2 + (b\sin\theta)^2} && \text{(B.23)}
\end{aligned}$$

If using $0 \le \theta < \pi$, then the arctan function is

$$\arctan\left(\frac{b}{a}\tan\theta\right) = \begin{cases} \operatorname{Arctan}\left(\frac{b}{a}\tan\theta\right) & \text{if } 0 \le \theta < \frac{\pi}{2} \\ \operatorname{Arctan}\left(\frac{b}{a}\tan\theta\right) + \pi & \text{if } \frac{\pi}{2} \le \theta < \pi \end{cases} \tag{B.24}$$

B.1.5 Convolution

Assume the function $h(x,y)$ being a 2D convolution of $f(x,y)$ and $g(x,y)$.

$$h(x,y) = f(x,y) ** g(x,y) = \int\int f(x_1,y_1)\; g(x-x_1, y-y_1)\; dx_1\; dy_1 \tag{B.25}$$

Then the Radon transform of $h(x,y)$ is given by

$$\begin{aligned}
\check{h}(\rho,\theta) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x_1,y_1)\; g(x-x_1,y-y_1)\; \delta(\rho - x\cos\theta - y\sin\theta)\; dx_1\; dy_1\; dx\; dy \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x_1,y_1) \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x-x_1,y-y_1)\; \delta(\rho - x\cos\theta - y\sin\theta)\; dx\; dy\; dx_1\; dy_1 \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x_1,y_1)\; \check{g}(\rho - x_1\cos\theta - y_1\sin\theta, \theta)\; dx_1\; dy_1 \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x_1,y_1) \int_{-\infty}^{\infty} \check{g}(\rho-\rho_1,\theta)\; \delta(\rho_1 - x_1\cos\theta - y_1\sin\theta,\theta)\; d\rho_1\; dx_1\; dy_1 \\
&= \int_{-\infty}^{\infty} \check{g}(\rho-\rho_1,\theta) \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x_1,y_1)\; \delta(\rho_1 - x_1\cos\theta - y_1\sin\theta,\theta)\; dx_1\; dy_1\; d\rho_1 \\
&= \int_{-\infty}^{\infty} \check{f}(\rho_1,\theta)\; \check{g}(\rho-\rho_1,\theta)\; d\rho_1 \qquad \text{(B.26)} \\
&= \check{f}(\rho,\theta) \;*\; \check{g}(\rho,\theta) \qquad \text{(B.27)}
\end{aligned}$$

The result is that the Radon transform of a two dimensional convolution is a one dimensional convolution of the Radon transformed functions with respect to ρ. The result is quite obvious from the Fourier Slice Theorem shown in Eq. 7.6. The two dimensional Fourier transformation of a convolution becomes a product. The product is still a product expressed in polar coordinates. The inverse one dimensional Radon transform of the product becomes a convolution in ρ.

B.2 The Shepp-Logan Phantom Brain

A 'famous' model of a brain is the Shepp Logan Phantom. This is a model based on ellipses, which can be Radon transformed analytically. The phantom can be used to test the numerical algorithms and evaluate the numerical errors quantitatively. Using the properties shown in Section B.1 a sum of ellipses can be transformed analytically, if a circle with radius one can be transformed, as shown in Fig. B.1. A unit circle is here defined as

$$g(x,y) = \begin{cases} 1 \text{ if } x^2+y^2 < 1 \\ 0 \text{ otherwise} \end{cases} \tag{B.28}$$

In this case the function is invariant to rotation around the origin of the coordinate system, thus the Radon transform does not depend on θ. The Radon transform can, e.g., be calculated for $\theta = 0$, as shown in Fig. B.2. Due to the constant excitation on the circle, the Radon transform is merely the length of the line crossing the circle.

From Fig. B.2 it is found that

$$\check{g}(\rho,\theta) = \begin{cases} 2\sqrt{1-\rho^2} \text{ if } \rho^2 < 1 \\ 0 \text{ otherwise} \end{cases} \tag{B.29}$$

Using Eq. B.29 and the properties shown in appendix B, it is found that a sum of ellipses can be transformed analytically. If the unit circle is scaled, rotated and shifted an ellipse is formed, shown in Fig. B.3.

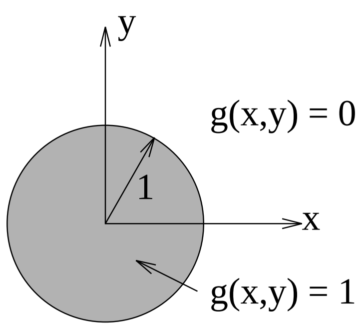

Figure B.1 The unit circle

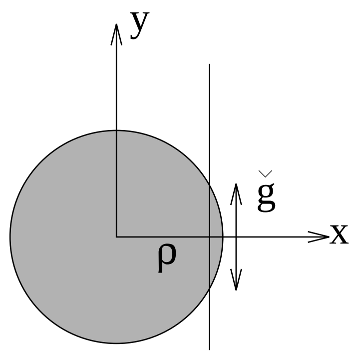

Figure B.2 Radon transformation for $\theta = 0$.

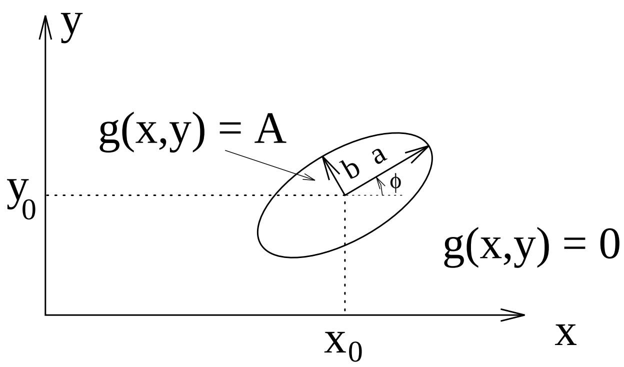

Figure B.3 An ellipse with the relevant parameters indicated

If the ellipse is called $g_{A,a,b,\phi,x_0,y_0}(x,y)$, then the Radon transform is given by

$$\check{g}_{A,a,b,\phi,x_0,y_0}(\rho,\theta) = \begin{cases} 2Aab\,\dfrac{\sqrt{\rho_0^2-(\rho-x_0\cos\theta-y_0\sin\theta)^2}}{\rho_0^2} & \text{if } |\rho-x_0\cos\theta-y_0\sin\theta|<\rho_0 \\ 0 & \text{otherwise} \end{cases} \tag{B.30}$$

$$\rho_0^2 = a^2\cos^2(\theta-\phi)+b^2\sin^2(\theta-\phi) \tag{B.31}$$

If Q ellipses are given with a set of parameters, then due to the linearity, the Radon transform is a sum

$$g(x,y) = \sum_{q=1}^{Q} g_{A_q,a_q,b_q,\phi_q,x_{0,q},y_{0,q}}(x,y) \;\Rightarrow \tag{B.32}$$

$$\check{g}(\rho,\theta) = \sum_{q=1}^{Q} \check{g}_{A_q,a_q,b_q,\phi_q,x_{0,q},y_{0,q}}(\rho,\theta) \tag{B.33}$$

The Shepp-Logan Phantom is a sum of ten ellipses with parameters, shown in Table B.1. Note that the ellipse amplitude A does not exactly match those of the original phantom. This choice gives a better contrast.

The resulting phantom brain is shown in Fig. B.4 and the corresponding Radon transform is shown in Fig. B.5.

A	a	b	x_0	y_0	ϕ
1.0	0.69	0.92	0.0	0.0	0^0
-0.8	0.6624	0.874	0.0	-0.0184	0^0
-0.2	0.11	0.31	0.22	0.0	-18^0
-0.2	0.16	0.41	-0.22	0.0	18^0
0.1	0.21	0.25	0.0	0.35	0^0
0.1	0.046	0.046	0.0	0.1	0^0
0.1	0.046	0.046	0.0	-0.1	0^0
0.1	0.046	0.023	-0.08	-0.605	0^0
0.1	0.023	0.023	0.0	-0.606	0^0
0.1	0.023	0.046	0.06	-0.605	0^0

Table B.1 Parameter settings for the Shepp Logan Phantom.

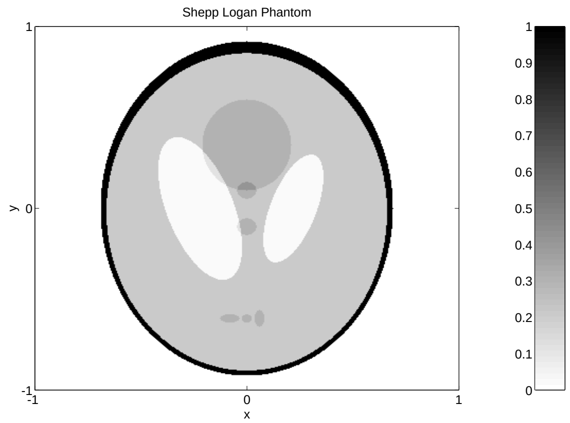

Figure B.4 The Shepp Logan Phantom brain.

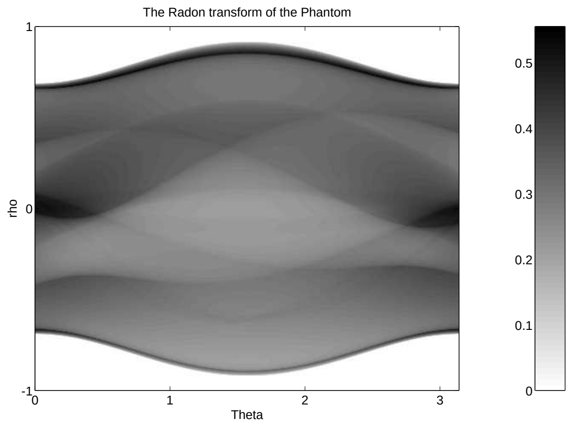

Figure B.5 The corresponding Radon transform.

B.3 Analytical Radon Transform of Primitives

During the project the normal Radon transform of several functions were derived analytically. The results are given in the following subsections. Using the linearity of the Radon transform and basic rules about translation, rotation and scaling rather complicated two dimensional functions can be modelled and the corresponding Radon transform can expressed on an analytical form. This provides a useful tool to generate test sets for testing of reconstruction algorithms. A program is available, where the theory is implemented and sampled versions of the functions and their Radon transform can be obtained. In [22] and Section C.1 are also shown how to use the program. The following paragraphs show which basic functions and their Radon transform, available in the program. The program is available from [10].

B.3.1 The Circular Disc

The normal Radon transform of the circular discnormal Radon transform was found in Eq. B.29. In Fig. B.6 the basic function is shown and Fig. B.7 shows the corresponding parameter domain. Due to the rotational symmetry of the disc, the Radon transform does not depend on the angle θ.

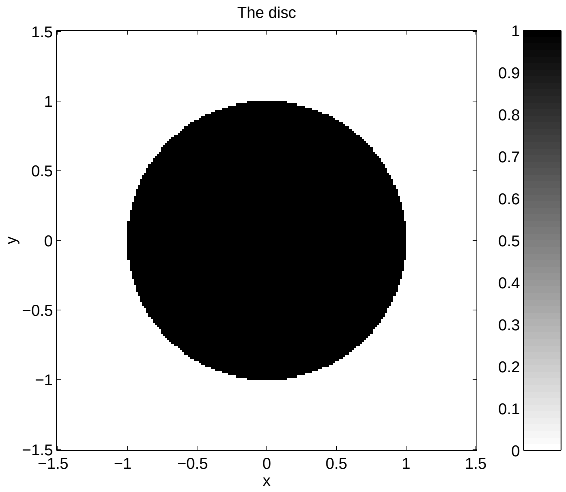

Figure B.6 The disc.

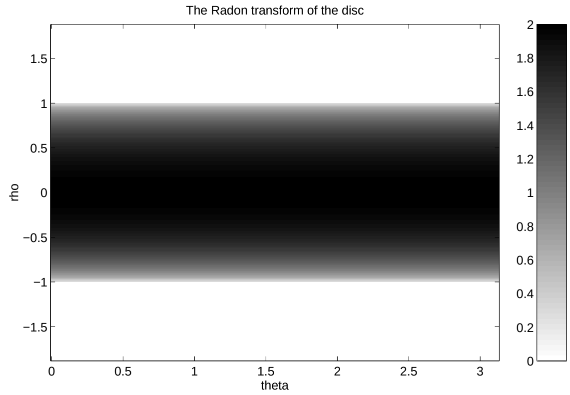

Figure B.7 The corresponding Radon transform.

B.3.2 The Square

Another very useful primitive is the square as shown in Fig. B.8. This function can be written as

$$g_s(x, y) = \mu(1 - |x|)\mu(1 - |y|) \tag{B.34}$$

where $\mu(\cdot)$ is the Hamilton step function. It is easy to see that the Radon transform of this primitive must exhibit several symmetries.

$$g_s(x, y) = g_s(|x|, |y|) \tag{B.35}$$

$$\check{g}_s(\rho, \theta) = \check{g}_s(\rho, \theta + p\frac{\pi}{2}) \tag{B.36}$$

$$= \check{g}_s(\rho, \frac{\pi}{2} - \theta) \tag{B.37}$$

where p is an integer. Eq. B.36 implies that the Radon transform is only needed in the interval $[0; \pi/2]$. Eq. B.37 further sharpens this demand to the interval $[0; \pi/4[$. Using the general feature $\check{g}(\rho, \theta) = \check{g}(-\rho, \theta + \pi)$ implies together with Eq. B.36 that only $\rho \geq 0$ must be investigated. It is also easy to find that

$$\check{g}_s(\rho, \theta) = 0 \text{ if } \rho > \sqrt{2} \tag{B.38}$$

The remaining case $\rho < \sqrt{2}$ is fairly easy due to the simple geometry. First are the intersections between the line $\rho = x\cos\theta + y\sin\theta$ and the boundaries found

$$y = -1 \Rightarrow x = x_{-1} = \frac{\rho}{\cos\theta} + \tan\theta \tag{B.39}$$

$$y = 1 \Rightarrow x = x_1 = \frac{\rho}{\cos\theta} - \tan\theta \tag{B.40}$$

$$x = 1 \Rightarrow y = y_1 = \frac{\rho}{\sin\theta} - \cot\theta \tag{B.41}$$

Three cases exist

$$x_1 > 0 \Rightarrow \check{g}(\rho, \theta) = 0 \tag{B.42}$$

$$x_1 < 1 \text{ and } x_{-1} < 1 \;\Rightarrow \tag{B.43}$$
$$\breve{g}(\rho,\theta) = \sqrt{2^2 + (x_1 - x_{-1})^2} \tag{B.44}$$
$$= \frac{2}{\cos\theta} \tag{B.45}$$

$$x_1 < 1 \text{ and } x_{-1} > 1 \;\Rightarrow \tag{B.46}$$
$$\breve{g}(\rho,\theta) = \sqrt{(1-x_1)^2 + (1-x_{-1})^2} \tag{B.47}$$

These formulas are computed in their respective region giving the total Radon transform of the square. In Fig. B.8 the square is shown and Fig. B.9 shows the corresponding Radon transform.

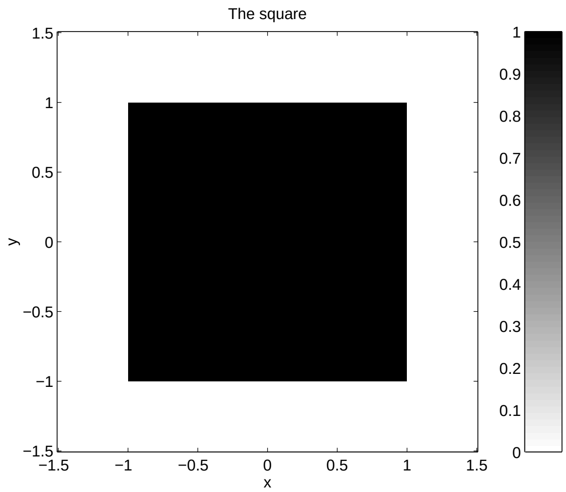

Figure B.8 The square.

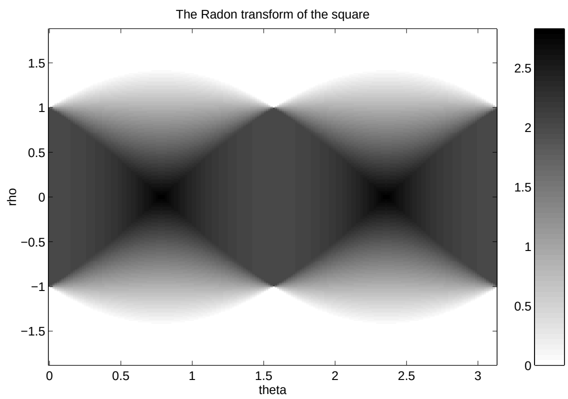

Figure B.9 The Radon transform of the square.

B.3.3 The Triangle

The following primitive is the triangle shown in Fig. B.10. This Radon transform of this primitive can also calculated on a analytical form. The first thing is that the rotational symmetry implies that

$$\breve{g}(\rho,\theta) = \breve{g}(\rho,-\theta) = \breve{g}\left(\rho, \theta + p\frac{2\pi}{3}\right) \tag{B.48}$$

where p is an integer. These relation together with $\breve{g}(\rho,\theta) = \breve{g}(-\rho,\theta+\pi)$ implies that it is only relevant to investigate $\rho \geq 0$ and $0 \leq \theta < \frac{\pi}{3}$.

Three lines are relevant

- $l_- : y = -\frac{1}{\sqrt{3}}(x-1)$, i.e., the upper slanted line of the triangle.
- $l_+ : y = \frac{1}{\sqrt{3}}(x-1)$, i.e., the lower slanted line of the triangle.
- $l : \rho = x\cos\theta + y\sin\theta$, i.e., the transformation line.

First is the intersection between l and l_+ is found, i.e.,

$$x_+ = \frac{\sin\theta + \sqrt{3}\rho}{\sin\theta + \sqrt{3}\cos\theta} \tag{B.49}$$

$$y_+ = \frac{1}{\sqrt{3}}(x_+ - 1) \tag{B.50}$$

and the intersection between l and l_-

$$x_- = \frac{\sin\theta - \sqrt{3}\rho}{\sin\theta - \sqrt{3}\cos\theta} \tag{B.51}$$

$$y_- = -\frac{1}{\sqrt{3}}(x_- - 1) \tag{B.52}$$

Another interesting intersection is between the vertical line $x = -\frac{1}{2}$ and the integration line. The corresponding $y = y_l$ is given by

$$y_l = \frac{2\rho + \cos\theta}{2\sin\theta} \tag{B.53}$$

It is easy to realize that $x_+ > 0$ implies that the integration line does not cross the triangle, i.e., $\breve{g}(\rho,\theta) = 0$.

The remaining cases all imply that the integration line intersects l_+

$$y_l < \frac{\sqrt{3}}{2} \;\Leftrightarrow\; \breve{g}(\rho,\theta) = \sqrt{\left(x_+ + \frac{1}{2}\right)^2 + (y_l - y_+)^2} \tag{B.54}$$

$$y_l > \frac{\sqrt{3}}{2} \;\Leftrightarrow\; \breve{g}(\rho,\theta) = \sqrt{(x_+ + x_-)^2 + (y_+ - y_-)^2} \tag{B.55}$$

These formulas are computed in their respective region giving the total Radon transform of the triangle. In Fig. B.10 the square is shown and Fig. B.11 shows the corresponding Radon transform.

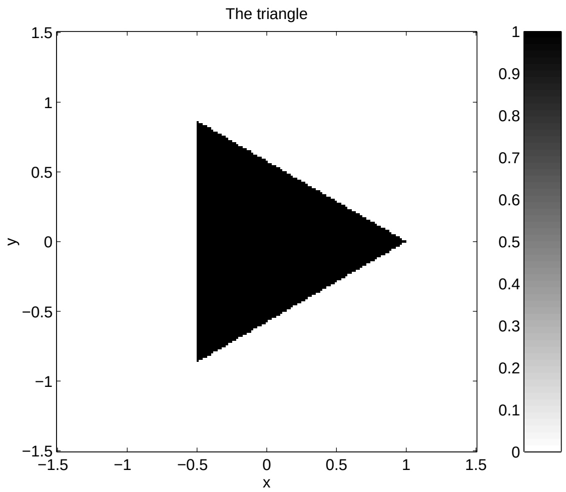

Figure B.10 The triangle.

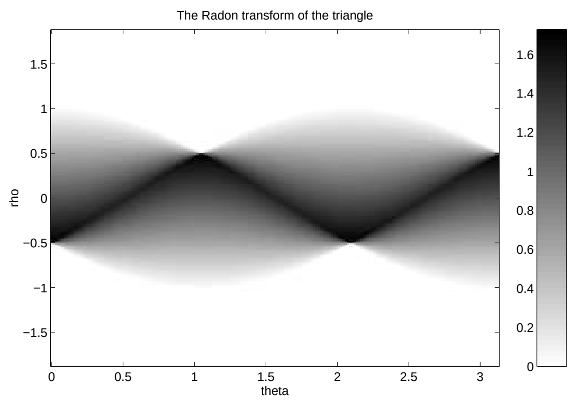

Figure B.11 The Radon transform of the triangle.

Using the presented set of primitives with proper scaling, rotation and translation as shown in the first part of this appendix, e.g., a “man” can be created as shown in Fig. B.12. The corresponding Radon transform is shown in Fig. B.13.

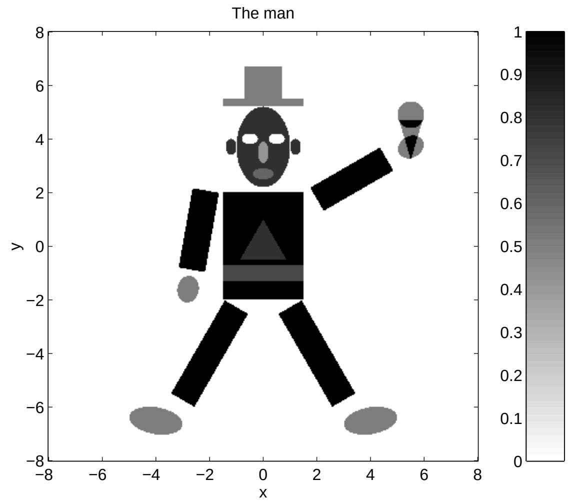

Figure B.12 The man.

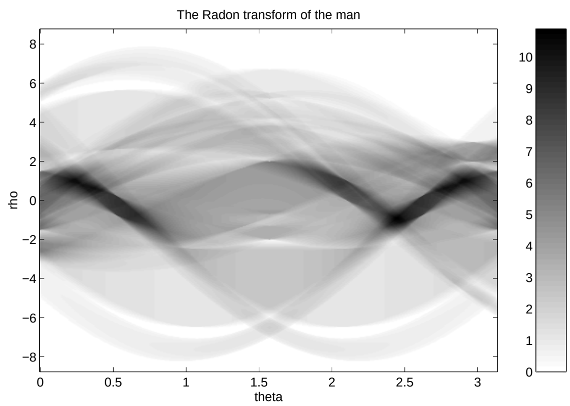

Figure B.13 The Radon transform of the Man.

B.3.4 The Gaussian Bell

The previous cases all concern finite objects with sharp edges. The next case is the infinite and soft varying Gaussian bell

$$g(x, y) = \exp(-x^2 - y^2) \tag{B.56}$$

Due to the rotational symmetry the Radon transform can be calculated at any angle, e.g., $\theta = 0$, where the integration reduces to

$$\begin{aligned} \breve{g}(\rho, \theta) &= \int_{-\infty}^{\infty} g(x = \rho, y)\ dy && \text{(B.57)} \\ &= \exp(-\rho^2) \int_{-\infty}^{\infty} \exp(-y^2)\ dy && \text{(B.58)} \\ &= \sqrt{\pi} \exp(-\rho^2) && \text{(B.59)} \end{aligned}$$

The Gaussian bell is shown in Fig. B.14 and the corresponding Radon transform given in Eq. B.59 is shown in Fig. B.15. Note that the bell is infinite, thus it cannot be bounded in the image- or the Radon-domain.

B.3.5 The Pyramid

The square can be used to simulate digital images using pixels with constant excitation. This is in fact a nearest neighbour approach. In some application, e.g., Radon transformation, the nearest neighbour approach introduces to much noise. In this case a linear interpolation strategy is very common. The discrete image is convoluted with a triangle in both directions, resulting in a pyramid-like structure. If normalizing the sampling distance to 1, the convolution kernel can be written as

$$g(x, y) = \begin{cases} (1 - |x|)(1 - |y|) & \text{for } |x| < 1, |y| < 1 \\ 0 & \text{otherwise} \end{cases} \tag{B.60}$$

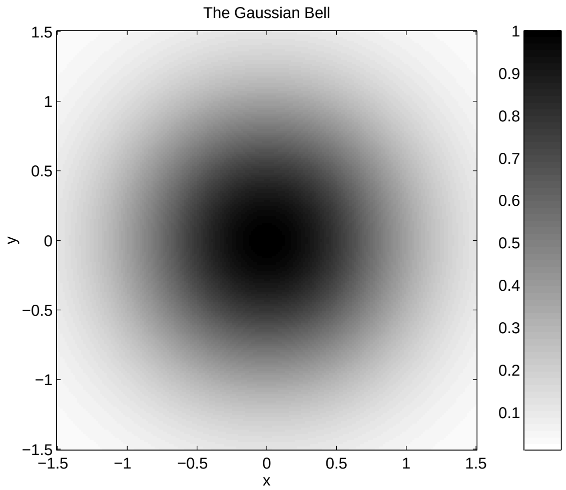

Figure B.14 The Gaussian Bell.

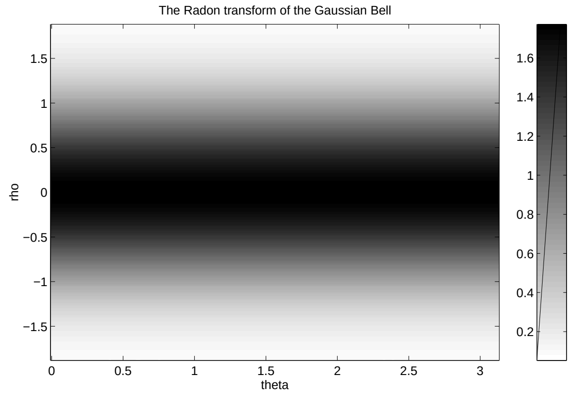

Figure B.15 The Radon transform of the Gaussian Bell.

The interpolated image can written as the integral

$$s_i(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\sum_m\sum_n s(m,n)\ \delta(x'-x_m)\ \delta(y'-y_n)\ g(x-x',y-y')\ dx'\ dy' \quad \text{(B.61)}$$

$$= \sum_m\sum_n s(m,n)\ g(x-x_m,y-y_n) \quad \text{(B.62)}$$

where $s(m,n)$ is the discrete image. Due to the linearity and translation property the Radon transform of the continuous function $s_i(x,y)$ can found if the Radon transform of the kernel $g(x,y)$ can be found. This will be done.

Due to the symmetry in the pyramid-like kernel it is easy to find that

$$\breve{g}(\rho,\theta) = \breve{g}(-\rho,\theta) = \breve{g}(\rho,\theta+p\pi/2) = \breve{g}(\rho,\pi/2-\theta) \quad \text{(B.63)}$$

It is also apparent that the Radon transform is zero if $\rho > \sqrt{2}$, i.e., the Radon transform is only needed in the interval $0 < \rho < \sqrt{2}$ and $0 \leq \theta < \pi/4$.

$$\breve{g}(\rho,\theta) = \frac{1}{|\cos\theta|}\int_{-\infty}^{\infty} g\left(\frac{\rho}{\cos\theta} - y\tan\theta, y\right) \quad \text{(B.64)}$$

$$= \frac{1}{|\cos\theta|}\int_{y^*}^{1} (\cos\theta - |\rho - y\sin\theta|)(1-|y|)\ dy \quad \text{(B.65)}$$

where

$$y^* = \max\{-1, y_1\} \quad \text{(B.66)}$$

and

$$y_1 = \frac{\rho - \cos\theta}{\sin\theta} \quad \text{(B.67)}$$

For test purposes it can used that $y_1 > 1$ implies that $\breve{g}(\rho,\theta) = 0$. The meaning of y_1 is indicated in Fig. B.16.

Yet another definition. It turns out that a vital parameter is

$$y_0 = \frac{\rho}{\sin\theta} \quad \text{(B.68)}$$

Solving the rather simple integrals in Eq. B.65 (it only requires good bookkeeping) the Radon transform can be derived. The result can be written as

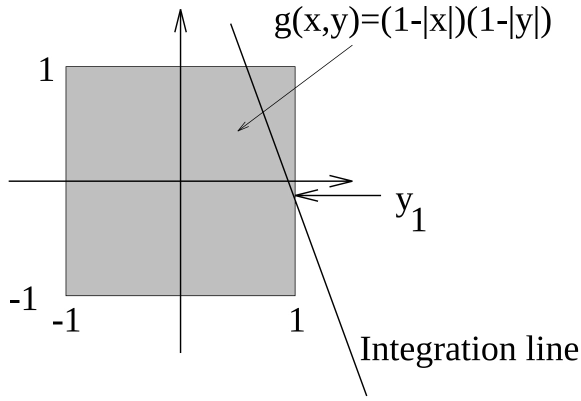

Figure B.16 The Pyramid.

$y_1 < -1$ and $y_0 > 1$

$$\breve{g}(\rho,\theta) = \frac{3\cos\theta - \rho}{\cos^2\theta} \tag{B.69}$$

$y_1 < -1$ and $y_0 < 1$

$$\breve{g}(\rho,\theta) = \frac{9\cos\theta - \sin\theta - 3\rho y_0 + \rho y_0^2}{3\cos^2\theta} \tag{B.70}$$

$-1 < y_1 < 0$ and $y_0 > 1$

$$\breve{g}(\rho,\theta) = \frac{\cos\theta(9 - 3y_1 + 5y_1^2) - 3\rho + 3\rho y_1 + \rho y_1^2 + \sin\theta}{6\cos^2\theta} \tag{B.71}$$

$-1 < y_1 < 0$ and $y_0 < 1$

$$\breve{g}(\rho,\theta) = \frac{\cos\theta(9 - 3y_1 + 5y_1^2) + \rho(3 - 3y_0 + 2y_0^2 + 3y_1 + y_1^2) - \sin\theta}{6\cos^2\theta} \tag{B.72}$$

$y_1 > 0$ and $y_0 > 1$

$$\breve{g}(\rho,\theta) = \frac{\cos\theta(9 - 3y_1 - 5y_1^2) + \rho(3y_1 - 3 - y_0 - y_1^2) + \sin\theta}{6\cos^2\theta} \tag{B.73}$$

$y_1 > 0$ and $y_0 < 1$

$$\breve{g}(\rho,\theta) = \frac{\cos\theta(9 - 3y_1 - 5y_1^2) + \rho(3 - 6y_0 + 3y_1 + 2y_0^2 - y_1^2) - \sin\theta}{6\cos^2\theta} \tag{B.74}$$

In Fig. B.17 the basic "pyramid"-function $g(x,y)$ is shown and finally Fig. B.18 shows the corresponding Radon transform.

Note that the Radon transform could also be calculated from the square using that the auto correlation function can be Radon transformed.

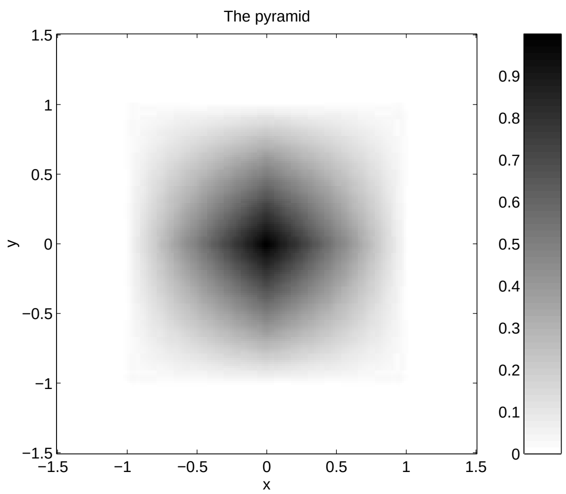

Figure B.17 The pyramid.

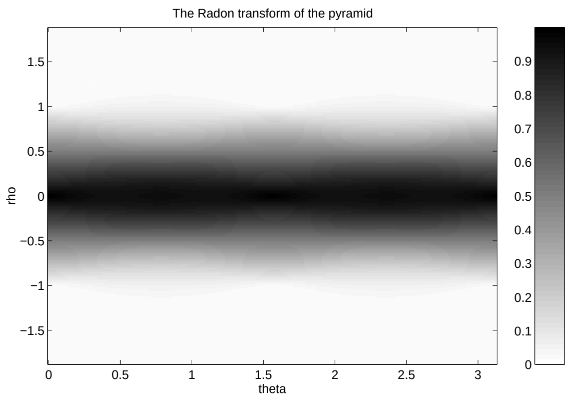

Figure B.18 The Radon transform of the pyramid.

Appendix C

Usage of the 2D Program Packages

Several software packages has been made for reconstruction of sinograms. The packages are provided for free, but protected by the GNU General Public License. All programs described in this appendix has been developed on a a Linux system and implemented in C. The program should compile directly on any GNU-C Unix system. The programs are available from [10].

In Section C.1 a program called "RadonAna" is described. The program is based on the derivation of the Radon transform of several well known and some new primitives, shown in Section B.3. The program can be used to generate pairs of analytically images and their Radon transform. The program can, e.g., be used to benchmark reconstruction packages.

In Section C.2 the usage of the program "iradon" is described. In the program several of the classical reconstruction techniques have been implemented.

In Section C.3 the iterative reconstruction program "it" is described. Here slow and fast versions of ART, EM and LSCG reconstruction have been implemented.

C.1 The Analytical Sinogram Program "RadonAna"

The program `RadonAna` can be used to generate Radon transform pairs. The basic idea is to specify a set of scaled, rotated, and shifted primitives from which the Radon transform is calculated and sampled in both the image domain and the Radon domain.

The program will produce two files `OutFileNamea.fif` and `OutFileNamer`.fif, where the file `OutFileName` is specified by the user, and the a-file is the image and the r-file is the radon space.

The program uses the following parameters

`OutFileName` [String] With this parameter the base file name is determined. If `OutFileName` is `man.fif` then the image is written in `mana.fif` and the sinogram in the file `manr.fif`.

`XSamples` [Integer] Specifies the number of image samples on the first axis. Should be an odd number.

`YSamples` [Integer] Specifies the number of image samples on the second axis. Should be an odd number. If not given the program uses `XSamples=YSamples`.

`DeltaX` [Float] Specifies the sampling distance of both axes in the image.

`Xmin` [Float] Specifies the minimum sample position in the image on the first axis. If not given the image is centered around the middle.

`Ymin` [Float] Specifies the minimum sample position in the image on the second axis. If not given the image is centered around the middle.

`RhoSamples` [Integer] Specifies the number of samples in the distance parameter ρ in the sinogram. Should be an odd number.

`ThetaSamples` [Integer] Specifies the number of samples in the angular parameter θ in the sinogram. The sinogram is sampled linearly from 0 to (approximately) π radians. The sampling distance in θ is π/`ThetaSamples`.

`DeltaRho` [Float] Specifies the sampling distance in ρ. The program will center the sampling points around 0.

`OverSamp` [Integer] This parameter specifies whether an oversampled image and sinogram should be generated or not, i.e., if `OverSamp`= 0. If, e.g., `OverSamp=2` an image is calculated with 5 times (2*`OverSamp`+1) the resolution in both parameters. This image is then averaged so the output value make a better approximation to the average value under the sample. This technique is also applied the the sinogram. It will reduce aliasing problems.

`GenerateImage` [Integer] If set to 1 the image is generated.

`GenerateRadon` [Integer] If set to 1 the sinogram is generated.

`NumberOfShapes` [Integer] Here the number of primitives are specified.

`DebugNiveau` [String] Log files is generated if this parameter is set to `_Debug`. No log-file and no screen output can be used with `DebugNiveau=_HardCore`.

The ordering of the parameters shown above are arbitrary. After these parameters a number of lines must follow specifying a scaling, rotation, and shifting parameters of each primitive. The line uses the following parameters in this order

`Shape`i [String] where i is 0 to `NumberOfShapes`-1.

`Type` [Integer] The type of primitive, where

- `1` A circle centered around (0,0) with radius 1 and uniform signal 1 on the disk.
- `2` A square centered around (0,0). $-1 < x < 1$ and $-1 < y < 1$. The signal is 0 on the disk.
- `3` A Gaussian bell $\exp(-x^2 - y^2)$.
- `4` An even sided triangle centered around $(0, 0)$. The triangle has corners $(1, 0)$, $(-\frac{1}{2}, \frac{\sqrt{3}}{2})$ and $(-\frac{1}{2}, -\frac{\sqrt{3}}{2})$.
- `5` The image and the Radon domain is con-terminated with white Gaussian noise. Note that in this case the noise term in the Radon domain is NOT the Radon transform of the noise in the image.
- `6` A "pyramid" $f(x, y) = (1 - |x|)(1 - |y|)$ for $-1 < x < 1$ and $-1 < y < 1$.

`a` [Float] Scaling of the primitive in the direction of the first axis.

`b` [Float] Scaling of the primitive in the direction of the second axis.

`x0` [Float] Shift of the primitive in direction of the first axis.

`y0` [Float] Shift of the primitive in direction of the second axis.

`phi` [Float] Rotation of the primitive measured in degrees.

`power` [Float] Scaling of the intensity of the primitive.

`Description` [String] Optional description of the primitive.

The program is executed with the initialization file or `.INI`-file as sole argument. A typical `.INI`-file could be the one used to generate the man. Note that comments can be used, such as the line above `Shape0`.

```
OutFileName=man.fif
XSamples=401          Number of X samples
DeltaX=0.04           Sampling Distance in X
RhoSamples=351        Number of rho samples
OverSamp=0            Half oversampling factor.
GenerateImage=1       Generates image if 1. Optional -> Generates if omitted
GenerateRadon=1       Generates sinogram if 1. Optional -> Generates if omitted
DeltaRho=0.05         Sampling distance in rho
ThetaSamples=300      Number of theta samples. Distributed from 0 to pi.
NumberOfShapes=22     Number of simple elements.
DebugNiveau=_DDebug

        type  a      b      x0    y0      phi power  description

Shape0=  1   1      1.5    0.0   3.7      0   0.8   head
Shape1=  1   0.4    0.2    0.0   2.7      0  -0.2   mouth
Shape2=  2   1.5    2.0    0.0   0.0      0   1.0   stomach
Shape3=  2   1.5    0.3    0.0  -1.0      0  -0.3   belt
Shape4=  2   1.5    0.5    3.3   2.5     30   1.0   right arm
Shape5=  2   1.5    0.5   -2.4   0.6   -100   1.0   left arm
Shape6=  4   1      1.0    0.0   0.0     90  -0.2   triangle on stomach
Shape7=  1   0.2    0.3   -1.2   3.7      0   0.8   left ear
Shape8=  1   0.2    0.3    1.2   3.7      0   0.8   right ear
Shape9=  2   1.5    0.15   0.0  5.35      0   0.5   lower part of hat
Shape10= 2   0.7    0.6    0.0   6.1      0   0.5   upper part of hat
Shape11= 1   0.3    0.2   -0.5   4.0      0  -0.8   left eye
Shape12= 1   0.3    0.2    0.5   4.0      0  -0.8   right eye
Shape13= 1   0.2    0.4    0.0   3.5      0  -0.4   nose
Shape14= 2   2.0    0.5    2.0  -4.0    -60   1.0   right leg
Shape15= 1   0.5    0.4    5.5   3.7     30   0.5   right hand
Shape16= 1   1      0.5    4.0  -6.5     10   0.5   left foot
Shape17= 2   2.0    0.5   -2.0  -4.0     60   1.0   left leg
Shape18= 1   0.5    0.4   -2.8  -1.6   -100   0.5   left hand
Shape19= 1   1      0.5   -4.0  -6.5    -10   0.5   right foot
Shape20= 4   1      0.5    5.5   4.2    -90   0.5   ice cream cone in right hand
Shape21= 1   0.5    0.5    5.5   4.9      0   0.5   only one ice cream cube!
```

C.2 The Direct Reconstruction Program "iradon"

In [59] the "iradon" package is described. Later additional features have been added, and here the basic possibilities are given.

The ordering of the following items which can be specified in the parameter file (.ini-file) are not important, but each require one line.

`OutFile` [String] Name of the file to write the reconstructed image (for the reconstruction functions). The program recognizes the following file extensions:

- `fif` Float image format which includes sampling parameters.
- `gif` The very popular graphics format.
- `dat` raw and outdated image format.
- `mat` Matlab files with one matrix.
- `analyze` GE specific file format.

`InFile` [String] Name of the file to read the sinogram image (for the reconstruction functions). The sinogram should be a fif-file and contain the sampling parameters.

`OrgFile` [String] (Optional) If the name of a fif-file is provided, the sampling parameters are used to create the output-file, and measures of misfit can be made between the OrgFile and the OutFile. This can be used to input the true solution and get error levels.

`Function` [String] The function the program should do. Currently supported functions are

- `FB` Filtered Backprojection.
- `BF` Filtering After Backprojection.
- `CNF` Central Slice. FFT based with Nearest Neighbour approximation.
- `CBF` Central Slice. FFT based with Bilinear Interpolation.
- `CNC` Central Slice. Chirp-z based with Nearest Neighbour approximation.
- `CBC` Central Slice. Chirp-z based with Bilinear Interpolation.
- `Forward` The InFile is forward Radon transformed into the OutFile.
- `Convert` Very handy function Convert Infile directly to OutFile (in another file-format).
- `Trace` Pick a trace of the InFile.
- `Info` Get statistics about the Infile.

`DebugLevel` [String] This parameter controls the level of output. The parameter is mixed and overrules with the one used in the Print-statements (extended printf defined in the program).

- `Normal` Standard level of output to screen and log file.
- `Debug` Almost all output is logged to screen and log file.
- `NoScreen` Screen output is disabled and log file level is ordinary.
- `NoLog` Log file output is disabled and screen level is ordinary.
- `HardCore` No information at all.

`Palette` [String] (Optional) Name of external colormap file to use when writing gif-images.

`InterPol` [Integer] (Optional) Interpolation level. Used by Filtered Backprojection to interpolate the sinogram. Default value is 1.

`FilterType` [String] (Optional) Filter choice in Filtered Backprojection. For all three choices see also `FilterCutoff`.

- `Ramp` Ordinary Ramp filter.
- `Hanning` The ramp filter is apodized with Hanning window.
- `Hamming` The ramp filter is apodized with Hamming window.

`FilterCutoff` [Float] (Optional) For Filtered Backprojection the frequency where the filter is set to zero. The number is relative to half of the sampling frequency, i.e., should be set between 0 and 1.

`SliceNumber` [Integer] (Optional) If the InFile is an analyze-file with a lot of slices, then this parameter will select the slice number.

`Xmin` [Float] The minimum x-position of the reconstructed image. Not needed if a fif-file is provided as OrgFile.

`Ymin` [Float] The minimum y-position of the reconstructed image. Not needed if a fif-file is provided as OrgFile.

`DeltaX` [Float] Sampling distance on the x-axis. Not needed if a fif-file is provided as OrgFile.

`DeltaY` [Float] Sampling distance on the y-axis. Not needed if a fif-file is provided as OrgFile.

`XSamples` [Integer] Number of samples on the x-axis. Not needed if a fif-file is provided as OrgFile.

`YSamples` [Integer] Number of samples on the y-axis. Not needed if a fif-file is provided as OrgFile.

Assume that the sinogram `mysino.fif` should be reconstructed into `rec.fif` using Filtered Backprojection with the same parameters as found in `myorg.fif`, then a possible parameter file `myrec.ini` could look like:

```
InFile=mysino.fif
OutFile=rec.fif
Function=FB
DebugLevel=Normal
InterPol=1
OrgFile=myorg.fif
```

C.3 The Fast Iterative Reconstruction Program "it"

The "it" program implements three of the major iterative reconstruction techniques: ART, EM, and LSCG, but with the basic framework provided virtually all iterative reconstruction algorithms based on matrix operations can easily be implemented. The program uses many of the same elements as "iradon" and provides several new for matrix operations relevant for iterative reconstruction.

`OutFileName` [String] Name of the file to write the reconstructed image. File formats are as described under "iradon".

`InFileName` [String] Name of the file where the sinogram image can be read.File formats are as described under "iradon".

`RefFileName` [String] If provided the reconstructed image will use sampling parameters from this image. Error measures can be made between the RefFile and the OutFile.

`StartFileName` [String] If provided the initial guess on the reconstructed image is taken from this image, else a constant solution will be assumed from start.

`Algorithm` [String] The iterative reconstruction method used: Currently three are available: ART, EM and CG (LSCG).

`UseFast` [Integer] If 1 then fast reconstruction is used, where the system matrix is stored using sparse techniques, else slower but memory efficient reconstruction is used.

`RadonKernel` [String] The type of kernel used to model the system matrix. Currently available are the following methods

- `NN` Two-level Nearest Neighbour approximation. (Memory consuming).
- `RNN` Ray driven Nearest Neighbour discrete Radon transform based (Very fast with small system matrix).
- `RL` Ray driven Linear Interpolation discrete Radon transform based (Fast with small system matrix).
- `P1` Method based on Radon transformation of square with pre-guidance (slow but good).
- `P1` Method based on Radon transformation of square with no pre-guidance (slower but better).
- `SINC` Sinc interpolation methods in the image and analytically Radon of that. (Very slow).

`IterationType` [Integer] For ART: If set to 1 a cyclical selection of the row index is used, and else a random selection is chosen.

`Iterations` [Integer] For EM and CG the number of iterations before the iteration ends. For ART the number of full iterations, i.e., the number of actual ART iterations is `Iterations` times the number of rows in the system matrix.

`SaveIterations` [Integer] (optional) If set to 1 the current solution will be saved after each iteration.

`LowestALevel` [Float] (optional) If fast reconstruction is used this the matrix elements are truncated to this level relative to the sampling distance of x, $\Delta x =$. If not set the value is 0.

`ConstrainMin` [Float] (optional) After each iteration the solution will forced up to this limit.

`ConstrainMax` [Float] (optional) After each iteration the solution will forced down to this limit. For both limit it is assumed that negative limits imply that the feature is not used.

`Alpha` [Float] (optional) For ART the initial update weight (if not specified then it is set to 1).

`Beta` [Float] (optional) For ART the multiplicative change to the weight factor, which should be less than one (if not specified then it is set to 1).

`Regularization` [Float] If set different from zero and using fast reconstruction, then the program will append rows to the system matrix with a simple Laplace operator. The parameter is used to control the degree of regularization.

`KernelFileName` [String] If using fast reconstruction the system matrix will be saved and restored with the extension <sif>. If a system matrix on the disk is incompatible with the sampling parameters, a new will be generated.

`SaveMatlab` [Integer] If the parameter is set to 1, then a sparse matrix is saved compatible with Matlab. The filename used will be KernelFileName.sia, where KernelFileName also should be specified.

`ThetaSamples` [Integer] Number of angular samples T in the sinogram, only needed if the sinogram file does not contain sampling information.

`ThetaMin` [Float] Start of the angular sampling (should be set to zero), only needed if the sinogram file does not contain sampling information.

`DeltaTheta` [Float] Angular sampling distance (should be set to π/ThetaSamples), only needed if the sinogram file does not contain sampling information.

`RhoSamples` [Integer] Number of samples in the sinogram R in the ρ-direction, only needed if the sinogram file does not contain sampling information.

`DeltaRho` [Integer] Sampling distance in ρ, i.e., $\Delta\rho$, only needed if the sinogram file does not contain sampling information.

`RhoMin` [Float] Start of sampling positions in ρ (should be set to $-\Delta\rho\frac{R-1}{2}$).

`Xmin` [Float] The minimum x-position of the reconstructed image. Not needed if a fif-file is provided as OrgFile.

`Ymin` [Float] The minimum y-position of the reconstructed image. Not needed if a fif-file is provided as OrgFile.

`DeltaX` [Float] Sampling distance on the x-axis. Not needed if a fif-file is provided as OrgFile.

`DeltaY` [Float] Sampling distance on the y-axis. Not needed if a fif-file is provided as OrgFile.

`XSamples` [Integer] Number of samples on the x-axis. Not needed if a fif-file is provided as OrgFile.

`YSamples` [Integer] Number of samples on the y-axis. Not needed if a fif-file is provided as OrgFile.

`DebugLevel` [String] This parameter controls the level of output. The parameter is mixed and overrules with the one used in the Print-statements (an expanded printf defined in the program).

`Normal` Standard level of output to screen and log file.

`Debug` Almost all output is logged to screen and log file.

`NoScreen` Screen output is disabled and log file level is ordinary.

`NoLog` Log file output is disabled and screen level is ordinary.

`HardCore` No information at all.

An example of a valid ini-file for "it" is shown below. The ini-file is used to reconstruct a sinogram `smallmanr.fif` into `smallman.EM.fif` with the same sampling parameters as contained in the file `smallmana.fif`. Here fast ART is used with a discrete Radon transform (linear interpolation) system matrix saved in `syssmall.sif`. Constraints are also used.

```
InFileName=smallmanr.fif
OutFileName=smallman.EM.fif
RefFileName=smallmana.fif
KernelFileName=syssmall.sif
Algorithm=ART
UseFast=1
RadonKernel=RL
Iterations=2
ConstrainMin=0
ConstrainMax=10
DebugLevel=Normal
```

Appendix D

The Three Dimensional Radon Transformation

The general form of the Radon transformation in N dimensions [14, 82] is given by

$$\breve{g}(\rho, \boldsymbol{\xi}) = \int_{-\infty}^{\infty} g(\boldsymbol{r})\ \delta(\rho - \boldsymbol{\xi} \cdot \boldsymbol{r})\ d\boldsymbol{r} \tag{D.1}$$

where $\boldsymbol{r}$ and $\boldsymbol{\xi} \in \mathbb{R}^N$ but $\boldsymbol{\xi}$ has only N-1 degrees of freedom, e.g., done by normalizing $|\boldsymbol{\xi}| = 1$. A common approach is to express the vector $\boldsymbol{\xi}$ in hyper-spherical coordinates [14]. In two dimensions $\boldsymbol{\xi} = (\cos\theta, \sin\theta)^T$ and in three dimensions the vector can be chosen to

$$\boldsymbol{\xi} = \begin{pmatrix} \cos\phi \sin\theta \\ \sin\phi \sin\theta \\ \cos\theta \end{pmatrix} \tag{D.2}$$

The fundamental body of the Radon transform is in two dimensions a line, cf. Eq. 2.4, and in three dimensions a plane described by the kernel of the Radon transform

$$\boldsymbol{\xi} \cdot \boldsymbol{r} = x \cos\phi \sin\theta + y \sin\phi \sin\theta + z \cos\theta \ \Rightarrow \tag{D.3}$$

$$\breve{g}(\rho, \theta, \phi) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(\boldsymbol{r})\ \delta(\rho - x \cos\phi \sin\theta - y \sin\phi \sin\theta - z \cos\theta)\ dx\ dy\ dz \tag{D.4}$$

Here it can be noted that the three dimensional Radon transform will transform a three dimensional signal $g(\boldsymbol{r})$ into a three dimensional parameter domain $\breve{g}(\rho, \theta, \phi)$.

The three dimensional Radon transform can be viewed as a convolution with a filter

$$\delta(\boldsymbol{\xi}) \tag{D.5}$$

where the $\boldsymbol{\xi}$ depends on the angles. Eq. D.5 shows that the Radon transform picks out a plane perpendicular to $\boldsymbol{\xi}$.

Like it was done for the two dimensional Radon transform the 3D Radon transform can also be written without the delta function, and here the integral is two dimensional. For sake of simplicity two other vectors orthogonal vectors are introduced, where $\boldsymbol{\xi}$, $\boldsymbol{\kappa}$, and $\boldsymbol{\gamma}$ are form a orthogonal basis.

$$\boldsymbol{\kappa} = \begin{pmatrix} -\sin\phi \\ \cos\phi \\ 0 \end{pmatrix} \qquad \boldsymbol{\gamma} = \begin{pmatrix} -\cos\phi \cos\theta \\ -\sin\phi \cos\theta \\ \sin\theta \end{pmatrix} \tag{D.6}$$

$$\breve{g}(\rho, \theta, \phi) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(r\boldsymbol{\kappa} + p\boldsymbol{\gamma} + \rho\boldsymbol{\xi})\ dr\ dp \tag{D.7}$$

In the following sections methods for inversion of the 3D Radon transform are derived.

D.1 The Three Dimensional Fourier Slice Theorem

The Fourier Slice Theorem can also be generalized to 3D. The inversion formula is based on Fourier transformation of Eq. D.1.

$$\begin{aligned}
\breve{G}(\nu,\boldsymbol{\xi}) &= \int_{-\infty}^{\infty} \breve{g}(\rho,\boldsymbol{\xi})\ e^{-j2\pi\rho\nu}\ d\rho && \text{(D.8)}\\
&= \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} g(\boldsymbol{r})\ \delta(\rho-\boldsymbol{\xi}\cdot\boldsymbol{r})\ d\boldsymbol{r}\right)\ e^{-j2\pi\rho\nu} d\rho && \text{(D.9)}\\
&= \int_{-\infty}^{\infty} g(\boldsymbol{r})\ e^{-j2\pi\nu\boldsymbol{\xi}\cdot\boldsymbol{r}}\ d\boldsymbol{r} && \text{(D.10)}
\end{aligned}$$

Defining the frequency parameters

$$\nu\boldsymbol{\xi} = \begin{pmatrix} u \\ v \\ w \end{pmatrix} \tag{D.11}$$

then the one dimensional Fourier transform of $\breve{g}$ can be recognized to the three dimensional Fourier transform of g.

$$\begin{aligned}
\breve{G}(\nu,\boldsymbol{\xi}) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y,z)\ e^{-j2\pi(xu+yv+zw)}\ dx\ dy\ dz \Leftrightarrow && \text{(D.12)}\\
g(\boldsymbol{r}) = g(x,y,z) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \breve{G}(\nu,\boldsymbol{\xi})\ e^{j2\pi(xu+yv+zw)}\ du\ dv\ dw && \text{(D.13)}
\end{aligned}$$

Note that the result is very similar to the two dimensional result and here the transformation is from a three dimensional domain into another three dimensional domain.

D.2 Filtered Backprojection in 3D

Filtered Backprojection can also be generalized into a 3D version. The inversion scheme is derived by expressing Eq. D.13 in spherical coordinates.

$$g(x,y,z) = g(\boldsymbol{r}) = \int_{\nu=0}^{\infty}\int_{\theta=0}^{\pi}\int_{\phi=0}^{2\pi} \nu^2 \sin\theta\ \breve{G}(\nu,\boldsymbol{\xi})\ e^{j2\pi\nu\boldsymbol{\xi}\cdot\boldsymbol{r}}\ d\phi\ d\theta\ d\nu \tag{D.14}$$

Now an inherent symmetry of spherical coordinates are used

$$\breve{G}(\nu,\boldsymbol{\xi}) = \breve{G}(\nu,\theta,\phi) = \breve{G}(-\nu,\pi-\theta,\phi-\pi) \tag{D.15}$$

which is inserted in Eq. D.14, and after a simple splitting of the ϕ integral, it is found that

$$g(\boldsymbol{r}) = \int_{\nu=-\infty}^{\infty}\int_{\theta=0}^{\pi}\int_{\phi=0}^{\pi} \nu^2 \sin\theta\ \breve{G}(\nu,\boldsymbol{\xi})\ e^{j2\pi\nu\boldsymbol{\xi}\cdot\boldsymbol{r}}\ d\phi\ d\theta\ d\nu \tag{D.16}$$

Now Eq. D.8 is inserted into Eq. D.16.

$$g(\boldsymbol{r}) = \int_{\nu=-\infty}^{\infty}\int_{\theta=0}^{\pi}\int_{\phi=0}^{\pi} \nu^2 \sin\theta\ \left(\int_{\rho=-\infty}^{\infty} \breve{g}(\rho,\theta,\phi)\ e^{-j2\pi\nu\rho}\ d\rho\right)\ e^{j2\pi\nu\boldsymbol{\xi}\cdot\boldsymbol{r}}\ d\phi\ d\theta\ d\nu \tag{D.17}$$

This equation can for convenience be written in two parts

$$\begin{aligned}
\bar{\breve{g}}(\rho,\theta,\phi) &= \int_{\nu=-\infty}^{\infty} \nu^2 \left(\int_{\rho=-\infty}^{\infty} \breve{g}(\tilde{\rho},\theta,\phi)\ e^{-j2\pi\nu\tilde{\rho}}\ d\tilde{\rho}\right) e^{j2\pi\nu\rho}\ d\nu && \text{(D.18)}\\
g(\boldsymbol{r}) &= \int_{\theta=0}^{\pi}\int_{\phi=0}^{\pi} \sin\theta\ \bar{g}(\boldsymbol{\xi}\cdot\boldsymbol{r},\theta,\phi)\ d\phi\ d\theta && \text{(D.19)}\\
&= \int_{\rho=-\infty}^{\infty}\int_{\theta=0}^{\pi}\int_{\phi=0}^{\pi} \sin\theta\ g^{\dagger}(\rho,\theta,\phi)\ \delta(\rho-\boldsymbol{\xi}\cdot\boldsymbol{r})\ d\phi\ d\theta\ d\rho && \text{(D.20)}
\end{aligned}$$

Note the similarity between the 2D Filtered Backprojection inversion scheme and the 3D version as shown in Eqs. D.18 and D.20. Naturally Eq. D.20 is named the backprojection operator. Note also that the filtering part of 3D Filtered Backprojection requires a quadratic filter ν^2.

These inversion schemes are generalizations of the ones shown for the two dimensional Radon transform, and implementation can be done by generalizing the techniques described in Chapter 8. It should be mentioned that [62] has a thorough description of the linogram method extended to three dimensions.

D.3 Connection between the 3D plane integrals and 3D line integrals

In this section the connection between the 3D line integrals (see Chapter 10) and the 3D plane integrals is established. This can be used to exploit the inverse 3D plane Radon transform, e.g., Filtered Backprojection shown in Eq. D.20 and D.18 to reconstruct a volume from line integrals and supplement the methods shown in Chapter 10.

Assume a plane with parameters ρ_p, θ_p, ϕ_p (index p for plane). From θ_p and ϕ_p an orthonormal basis can be found

$$\boldsymbol{\xi} = \begin{pmatrix} \sin\theta_p \cos\phi_p \\ \sin\theta_p \sin\phi_p \\ \cos\theta_p \end{pmatrix} \quad \boldsymbol{\zeta} = \begin{pmatrix} -\sin\phi_p \\ \cos\phi_p \\ 0 \end{pmatrix} \quad \boldsymbol{\eta} = \begin{pmatrix} \cos\theta_p \cos\phi_p \\ \cos\theta_p \sin\phi_p \\ -\sin\theta_p \end{pmatrix} \tag{D.21}$$

Now all lines in the plane (normal to $\boldsymbol{\xi}$) is be found. The direction vector $\boldsymbol{\tau}$ of the lines can in general be written as

$$\boldsymbol{\tau} = \boldsymbol{\eta} \cos\omega + \boldsymbol{\zeta} \sin\omega \tag{D.22}$$

The vectors are illustrated in Fig. D.1.

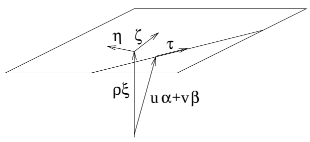

Figure D.1 The plane with normal vector $\boldsymbol{\xi}$. The line $\boldsymbol{r} = s\boldsymbol{\tau} + u\boldsymbol{\alpha} + v\boldsymbol{\beta}$ lies in the plane.

Another useful vector is $\boldsymbol{\chi}$, defined as the vector product between $\boldsymbol{\tau}$ and $\boldsymbol{\xi}$. Using that $\boldsymbol{\xi}$, $\boldsymbol{\eta}$ and $\boldsymbol{\zeta}$ are orthonormal vectors imply that

$$\boldsymbol{\chi} = \boldsymbol{\eta} \sin\omega - \boldsymbol{\zeta} \cos\omega \tag{D.23}$$

hence the lines can be described by

$$\boldsymbol{r} = s\boldsymbol{\tau} + \boldsymbol{r}_0 \tag{D.24}$$

where s is the free parameter and $\boldsymbol{r}_0$ is the base point of the lines, resolved as shown in Section 10.1. Using the symbols defined there gives

$$\boldsymbol{r}_0 = u\boldsymbol{\alpha} + v\boldsymbol{\beta} \tag{D.25}$$

Using the basis vectors shown in Fig. D.1 the plane integral $\breve{g}(\rho, \boldsymbol{\xi})$ can be expressed as

$$\breve{g}_\omega(\rho, \boldsymbol{\xi}) = \int_{r=-\infty}^{\infty} \int_{t=-\infty}^{\infty} g(\rho\boldsymbol{\xi} + r\boldsymbol{\chi} + t\boldsymbol{\tau})\ dt\ dr \tag{D.26}$$

$$= \int_{r=-\infty}^{\infty} \breve{g}(\theta_l, \phi_l, u, v)\ dr \tag{D.27}$$

where θ_l and ϕ_l (The index l stands for line) are the angles corresponding to $\boldsymbol{\tau}$ found from Eq. 10.2. The parameters u and v are found from

$$\rho\boldsymbol{\xi} + r\boldsymbol{\chi} = u\boldsymbol{\alpha} + v\boldsymbol{\beta} \tag{D.28}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are found from the angles θ_l and ϕ_l.

The result in Eq. D.27 depends on the angle ω. The final estimate of the plane integral may be set to a weighted integral

$$\breve{g}(\rho, \boldsymbol{\xi}) = \frac{1}{\pi} \int_{\omega=0}^{\pi} q(\omega)\ \breve{g}_\omega(\rho, \boldsymbol{\xi})\ d\omega \tag{D.29}$$

$$= \frac{1}{\pi} \int_{\omega=0}^{\pi} q(\omega) \int_{r=-\infty}^{\infty} \breve{g}(\theta_l, \phi_l, \boldsymbol{\alpha} \cdot (\rho\boldsymbol{\xi} + r\boldsymbol{\chi}), \boldsymbol{\beta} \cdot (\rho\boldsymbol{\xi} + r\boldsymbol{\chi}))\ dr\ d\omega \tag{D.30}$$

where $q(\omega)$ is a weight designed after the available data. In general it must fulfill

$$\int_{\omega=0}^{\pi} q(\omega)\ d\omega = \pi \tag{D.31}$$

Appendix E

Properties of the 3D line Radon Transform

From the 3D line Radon transform shown in Eq. E.1 several properties are derived.

E.1 Basic Properties of the 3D line Radon Transform

For sake of clarify the 3D line Radon transform given in Eq. 10.5, is repeated here.

$$\breve{g}(\theta, \phi, u, v) = \int_{-\infty}^{\infty} g(s\boldsymbol{\tau} + u\boldsymbol{\alpha} + v\boldsymbol{\beta}) \, ds \tag{E.1}$$

E.1.1 Linearity

The first crucial property is linearity of the transform.

$$g(\boldsymbol{r}) = \sum_i \kappa_i \; g_i(\boldsymbol{r}) \;\; \Rightarrow \;\; \breve{g}(\theta, \phi, u, v) = \sum_i \kappa_i \; \breve{g}_i(\theta, \phi, u, v) \tag{E.2}$$

where the set of κ_i are arbitrary constants.

E.1.2 Translation

It is assumed that

$$h(\boldsymbol{r}) \; = \; g(\boldsymbol{r} - \boldsymbol{r_0}) \;\; \Rightarrow \tag{E.3}$$

$$\breve{h}(\theta, \phi, u, v) \; = \; \int_{-\infty}^{\infty} g(s\boldsymbol{\tau} + u\boldsymbol{\alpha} + v\boldsymbol{\beta} - \boldsymbol{r_0}) \; ds \tag{E.4}$$

Here the translation $\boldsymbol{r_0}$ are resolved after the basis vectors, i.e.,

$$\boldsymbol{r_0} = s_0\boldsymbol{\tau} + u_0\boldsymbol{\alpha} + v_0\boldsymbol{\beta} \tag{E.5}$$

This can be done in an unambiguous way

$$s_0 = \boldsymbol{r_0} \cdot \boldsymbol{\tau} \quad u_0 = \boldsymbol{r_0} \cdot \boldsymbol{\alpha} \quad v_0 = \boldsymbol{r_0} \cdot \boldsymbol{\beta} \tag{E.6}$$

This is inserted in Eq. E.4.

$$\check{h}(\theta,\phi,u,v) = \int_{-\infty}^{\infty} g((s-s_0)\boldsymbol{\tau} + (u-u_0)\boldsymbol{\alpha} + (v-v_0)\boldsymbol{\beta})\ ds \tag{E.7}$$

$$= \int_{-\infty}^{\infty} g(\tilde{s}\boldsymbol{\tau} + (u-u_0)\boldsymbol{\alpha} + (v-v_0)\boldsymbol{\beta})\ d\tilde{s} \Rightarrow \tag{E.8}$$

$$\check{h}(\theta,\phi,u,v) = \breve{g}(\theta,\phi,u-\boldsymbol{r_0}\cdot\boldsymbol{\alpha}, v-\boldsymbol{r_0}\cdot\boldsymbol{\beta}) \tag{E.9}$$

Note that the angular parameters θ and ϕ are not changed.

E.1.3 Rotation and Scaling

Rotation and scaling can be controlled by means of a general transformation matrix $\boldsymbol{A}$

$$h(\boldsymbol{r}) = g(\boldsymbol{A}\boldsymbol{r}) \tag{E.10}$$

The diagonal elements of the $3 * 3$ matrix control the individual scaling and the off-diagonal elements are controlling the rotation. The corresponding Radon transform is given by

$$\check{h}(\theta,\phi,u,v) = \int_{-\infty}^{\infty} g(\boldsymbol{A}(s\boldsymbol{\tau} + u\boldsymbol{\alpha} + v\boldsymbol{\beta})\ ds \tag{E.11}$$

$$= \int_{-\infty}^{\infty} g(s\boldsymbol{A}\boldsymbol{\tau} + u\boldsymbol{A}\boldsymbol{\alpha} + v\boldsymbol{A}\boldsymbol{\beta})\ ds \tag{E.12}$$

This will in general alter the directional vector $\boldsymbol{\tau}$ of the line, i.e., a new direction vector is needed

$$\boldsymbol{A}\boldsymbol{\tau} = \gamma\tilde{\boldsymbol{\tau}} \quad |\tilde{\boldsymbol{\tau}}| = 1 \quad \gamma \geq 0 \tag{E.13}$$

The vector $\tilde{\boldsymbol{\tau}}$ defines cf. Eq. 10.2 two corresponding angles $\tilde{\theta}$ and $\tilde{\phi}$, which again defines the two remaining vectors $\tilde{\boldsymbol{\alpha}}$ and $\tilde{\boldsymbol{\beta}}$, cf. Eq. 10.4. Now Eq. E.13 is inserted into Eq. E.12.

$$\check{h}(\theta,\phi,u,v) = \frac{1}{\gamma}\int_{-\infty}^{\infty} g\left(s\tilde{\boldsymbol{\tau}} + \boldsymbol{A}\left(u\boldsymbol{\alpha} + v\boldsymbol{\beta}\right)\right)\ ds \tag{E.14}$$

The next step is to resolve the line offset parameters after the three new tilde vectors, which again can be done unambiguously.

$$\boldsymbol{A}\left(u\boldsymbol{\alpha} + v\boldsymbol{\beta}\right) = s_0\tilde{\boldsymbol{\tau}} + u_0\tilde{\boldsymbol{\alpha}} + v_0\tilde{\boldsymbol{\beta}} \tag{E.15}$$

This implies that the 3D line Radon transform is given by

$$\check{h}(\theta,\phi,u,v) = \frac{1}{\gamma}\int_{-\infty}^{\infty} g\left((s+s_0)\tilde{\boldsymbol{\tau}} + u_0\tilde{\boldsymbol{\alpha}} + v_0\tilde{\boldsymbol{\beta}}\right)\ ds \tag{E.16}$$

$$= \frac{1}{\gamma}\int_{-\infty}^{\infty} g\left(\tilde{s}\tilde{\boldsymbol{\tau}} + u_0\tilde{\boldsymbol{\alpha}} + v_0\tilde{\boldsymbol{\beta}}\right)\ d\tilde{s} \tag{E.17}$$

$$= \frac{1}{\gamma}\breve{g}(\tilde{\theta},\tilde{\phi},u_0,v_0) \tag{E.18}$$

where the four tilde parameters are given by

$$\tilde{\boldsymbol{\tau}} = \begin{pmatrix} \cos\tilde{\phi}\cos\tilde{\theta} \\ \cos\tilde{\phi}\sin\tilde{\theta} \\ \sin\tilde{\phi} \end{pmatrix} = \frac{1}{|\boldsymbol{A}\boldsymbol{\tau}|}\boldsymbol{A}\boldsymbol{\tau}, \quad \tilde{\boldsymbol{\alpha}} = \begin{pmatrix} -\sin\tilde{\theta} \\ \cos\tilde{\theta} \\ 0 \end{pmatrix} \text{ and } \tilde{\boldsymbol{\beta}} = \begin{pmatrix} -\sin\tilde{\phi}\cos\tilde{\theta} \\ -\sin\tilde{\phi}\sin\tilde{\theta} \\ \cos\tilde{\phi} \end{pmatrix} \tag{E.19}$$

$$\gamma = |\boldsymbol{A}\boldsymbol{\tau}|, \quad u_0 = \tilde{\boldsymbol{\alpha}}\cdot\boldsymbol{A}\cdot(\boldsymbol{\alpha}u + \boldsymbol{\beta}v), \quad v_0 = \tilde{\boldsymbol{\beta}}\cdot\boldsymbol{A}\cdot(\boldsymbol{\alpha}u + \boldsymbol{\beta}v) \tag{E.20}$$

The matrix $\boldsymbol{A}$ can be partitioned in three rotational matrices and one scaling matrix

$$\boldsymbol{A}_z \boldsymbol{A}_y \boldsymbol{A}_x \boldsymbol{S} \tag{E.21}$$

where $\boldsymbol{A}_x$ rotates in the $y-z$ coordinates, e.g.

$$\boldsymbol{A}_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi_x & \sin\psi_x \\ 0 & -\sin\psi_x & \cos\psi_x \end{pmatrix} \tag{E.22}$$

and $\boldsymbol{A}_y$ rotates in the $x-z$ coordinates, e.g.

$$\boldsymbol{A}_y = \begin{pmatrix} \cos\psi_y & 0 & \sin\psi_y \\ 0 & 1 & 0 \\ -\sin\psi_y & 0 & \cos\psi_y \end{pmatrix} \tag{E.23}$$

and finally $\boldsymbol{A}_z$ rotates in the $x-y$ coordinates, e.g.

$$\boldsymbol{A}_z = \begin{pmatrix} \cos\psi_z & \sin\psi_z & 0 \\ -\sin\psi_z & \cos\psi_z & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{E.24}$$

The scaling matrix $\boldsymbol{S}$ is here chosen to be the last in Eq. E.21 because the interpretation of the scaling is by far easier when applied directly on the base function $g(\boldsymbol{r})$. The matrix can be chosen to

$$\boldsymbol{S} = \begin{pmatrix} \frac{1}{S_x} & 0 & 0 \\ 0 & \frac{1}{S_y} & 0 \\ 0 & 0 & \frac{1}{S_z} \end{pmatrix} \tag{E.25}$$

The scaling parameters in the diagonal is chosen so they directly relate to length in the new function $h(\boldsymbol{r})$. The total rotation and scaling matrix defined in Eq. E.21 multiplied together becomes

$$\boldsymbol{A} = \begin{pmatrix} \dfrac{\cos\psi_x\cos\psi_y}{S_x} & \dfrac{\cos\psi_y\sin\psi_x}{S_y} & \dfrac{\sin\psi_y}{S_z} \\ -\dfrac{\cos\psi_z\sin\psi_x+\cos\psi_x\sin\psi_y\sin\psi_z}{S_x} & \dfrac{\cos\psi_x\cos\psi_z-\sin\psi_x\sin\psi_y\sin\psi_z}{S_y} & \dfrac{\cos\psi_y\sin\psi_z}{S_z} \\ \dfrac{\sin\psi_x\sin\psi_z-\cos\psi_x\cos\psi_z\sin\psi_y}{S_x} & -\dfrac{\cos\psi_z\sin\psi_x\sin\psi_y+\cos\psi_x\sin\psi_z}{S_y} & \dfrac{\cos\psi_y\cos\psi_z}{S_z} \end{pmatrix} \tag{E.26}$$

E.2 Analytical Radon Transformation of Primitives

Along with the basic rules of the 3D line Radon transform it is very useful to have a set of base 3D functions and their corresponding Radon transform. In the following the Radon transform of the unit ball and a Gaussian bell are derived. This can be used to produce more complex functions written as a weighted sum of shifted, rotated and scaled primitives.

E.2.1 The ball

The first primitive is the unit ball, i.e.,

$$g_{\text{ball}}(x, y, z) = \begin{cases} 1 \text{ for } x^2 + y^2 + z^2 < 1 \\ 0 \text{ for } x^2 + y^2 + z^2 \geq 1 \end{cases} \tag{E.27}$$

Due to the rotational symmetry of the primitive the corresponding Radon transform cannot depend on the two angular parameters θ and ϕ, thus the Radon transform is derived in a rotated coordinate system with $\boldsymbol{\tau}$ lying along the x-axis and the vector $\boldsymbol{r_0} = u\boldsymbol{\alpha} + v\boldsymbol{\beta}$ along the y-axis. As illustrated in Fig. E.1 is the Radon transform merely the distance between the two intersection points between the line and the ball.

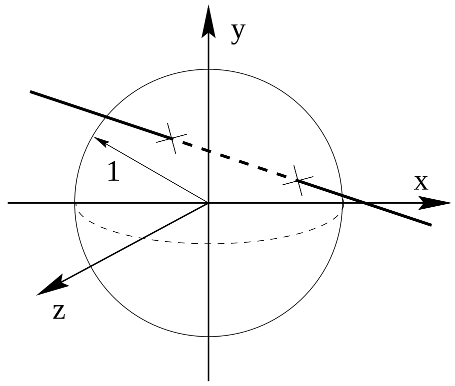

Figure E.1 The unit ball with radius 1 and the Radon transform is the length between the two intersection points between the line and the surface of the ball.

The length of $\boldsymbol{r_0}$ is $\sqrt{u^2 + v^2}$, which implies that the Radon transform can be calculated as

$$\breve{g}_{\text{ball}}(\theta, \phi, u, v) = \int_{-\infty}^{\infty} g_{\text{ball}}\left(s, \sqrt{u^2 + v^2}, 0\right) \, ds \;\Rightarrow \tag{E.28}$$

$$\breve{g}_{\text{ball}}(\theta, \phi, u, v) = \begin{cases} 2\sqrt{1 - u^2 - v^2} & \text{for } u^2 + v^2 < 1 \\ 0 & \text{for } u^2 + v^2 \geq 1 \end{cases} \tag{E.29}$$

E.2.2 The Gaussian bell

The Gaussian bell is another primitive, where the Radon transform can be found analytically.

$$g_{\text{gauss}}(x, y, z) = \exp(-x^2 - y^2 - z^2) \tag{E.30}$$

Again the function has rotational symmetry, hence the Radon transform does not depend on the angular parameters θ and ϕ, and the Radon transform can, e.g., be derived as

$$\breve{g}_{\text{gauss}}(\theta, \phi, u, v) = \int_{-\infty}^{\infty} e^{-s^2 - u^2 - v^2} \, ds \;\Rightarrow\; \breve{g}_{\text{gauss}}(\theta, \phi, u, v) = \sqrt{\pi} e^{-u^2 - v^2} \tag{E.31}$$

Appendix F

Usage of the 3D Reconstruction Tools

A software packages has been made for 3D reconstruction of sinograms (line integrals), as shown in Chapter 10. The package are provided for free, but protected by the GNU General Public License. The programs described in this appendix has been developed on a a Linux system and implemented in C. The program should compile directly on any GNU-C Unix system. Furthermore will the program compile on a SGI Onyx, and parallel options, `#pragma XXX`, have been added to take advantage of the parallel possibilities of Iris Power C. The package is available from [10].

In Section F.1 is the usage of the reconstruction program “Recon3D” described. As mentioned in Section 10.7 the program is intended for testing of different 3D reconstruction algorithms, and no Kinahan & Rogers reprojection step is currently included.

In Section F.2 is the program “3D_RadonAna” described, which can generate a volume and the corresponding four-dimensional sinogram from a set of scaled, translated, and rotated primitives.

F.1 The 3D Reconstruction Program “Recon3D”

The program “Recon3D” uses the following parameters

`DebugLevel` [String] This parameter controls the level of output. The parameter is mixed and overrules with the one used in the Print-statements (extended printf defined in the program).

- `Normal` Standard level of output to screen and log file.
- `Debug` Almost all output is logged to screen and log file.
- `NoScreen` Screen output is disabled and log file level is ordinary.
- `NoLog` Log file output is disabled and screen level is ordinary.
- `HardCore` No information at all.

`WorkDir` [String] Directory where the data-files are stored. The style `../signals` is valid.

`Sinogram` [String] Name of the file containing the 4D sinogram. The file must have extension `.f4f` (binary format) or `.a4f` (ASCII format).

`OrgFile` [String] (optional) Name of a volume specifying the sampling parameters of the reconstructed volume. If given, then error measures will be computed with respect to this volume.

`Volume` [String] Name of the volume to be reconstructed. The file must have extension `.f3f` (binary format) or `.a3f` (ASCII format).

`Function` [String] Parameter specifying the action to be used.

- `FB` Direct reconstruction using Filtered Backprojection. Result in the `Volume`.
- `FAB` Direct reconstruction using Filtering after Backprojection. Result in the `Volume`.
- `CS` Direct reconstruction using Fourier Slice Theorem (Experimental). Result in the `Volume`.
- `Forward` Radon transform of `Volume` into the `Sinogram`.
- `MART` Iterative reconstruction using MART. Result in the `Volume`.
- `ART` Iterative reconstruction using ART. Result in the `Volume`.
- `EM` Iterative reconstruction using EM. Result in the `Volume`.

`Iterations` [Integer] For EM the number of iterations before the iteration ends. For ART and MART the number of full iterations, i.e., the number of actual ART iterations is `Iterations` times the number of rows in the system matrix.

`Iterative_Par_1` [Float] In ART and MART the initial weight parameter λ.

`Iterative_Par_2` [Float] In ART and MART the final weight parameter λ.

`Select_Par_1` [Integer] In Fourier Slice Reconstruction a value of 0 will choose a nearest neighbour interpolation in the spectrum, and a value of 1 implies that tri-linear interpolation is used.

`Xmin` [Float] The minimum value of x in the volume. Only needed if `OrgFile` is not specified.

`Ymin` [Float] The minimum value of y in the volume. Only needed if `OrgFile` is not specified.

`Zmin` [Float] The minimum value of z in the volume. Only needed if `OrgFile` is not specified.

`DeltaX` [Float] The sampling interval of x in the volume. Only needed if `OrgFile` is not specified.

`DeltaY` [Float] The sampling interval of y in the volume. Only needed if `OrgFile` is not specified.

`DeltaZ` [Float] The sampling interval of z in the volume. Only needed if `OrgFile` is not specified.

`XSamples` [Integer] The number of samples of x in the volume. Only needed if `OrgFile` is not specified.

`YSamples` [Integer] The number of samples of y in the volume. Only needed if `OrgFile` is not specified.

`ZSamples` [Integer] The number of samples of z in the volume. Only needed if `OrgFile` is not specified.

An example of a valid ini-file for "Recon3D" is shown below. The ini-file is used to reconstruct a sinogram `../signals/Mickey_Sino.f4f` into `Mickey_Out.f3f` with the same sampling parameters as contained in the file `Mickey_Vol.f3f`. Here Filtered Backprojection is used.

```
WorkDir=../signals
Sinogram=Mickey_Sino.f4f
Volume=Mickey_Out.f3f
OrgFile=Mickey_Vol.f3f
Function=FB
DebugLevel=Normal
Iterations=30
```

F.2 The Analytical Sinogram Program "3D_RadonAna"

The program "3D_RadonAna" uses the following parameters

`DebugLevel` [String] This parameter controls the level of output. The parameter is mixed and overrules with the one used in the Print-statements (extended printf defined in the program).

- `Normal` Standard level of output to screen and log file.
- `Debug` Almost all output is logged to screen and log file.
- `NoScreen` Screen output is disabled and log file level is ordinary.
- `NoLog` Log file output is disabled and screen level is ordinary.
- `HardCore` No information at all.

`OutFileName` [String] First part of the name to output. Two files can be written. The volume in `OutFileName_Vol.f3f` and the sinogram in `OutFileName_Sino.f4f`. Here the style `../signals/MickeyBig` can be used.

`Xmin` [Float] The minimum value of x in the volume.

`Ymin` [Float] (optional) The minimum value of y in the volume. If not specified `Xmin` will be used.

`Zmin` [Float] (optional) The minimum value of z in the volume. If not specified `Xmin` will be used.

`DeltaX` [Float] The sampling interval of x in the volume.

`DeltaY` [Float] (optional) The sampling interval of x in the volume. If not specified the valued of `DeltaX` will be.

`DeltaZ` [Float] (optional) The sampling interval of z in the volume. If not specified the valued of `DeltaX` will be.

`XSamples` [Integer] The number of samples of x in the volume.

`YSamples` [Integer] (optional) The number of samples of y in the volume. If not specified `XSamples` will be used.

`ZSamples` [Integer] (optional) The number of samples of z in the volume. If not specified `XSamples` will be used.

`USamples` [Integer] The number of samples of u in the sinogram.

`VSamples` [Integer] The number of samples of v in the sinogram. If not specified `USamples` will be.

`DeltaU` [Float] The sampling interval of u in the sinogram.

`DeltaV` [Float] The sampling interval of v in the sinogram. If not specified `DeltaU` will be.

`PhiSamples` [Integer] The number of samples of ϕ in the sinogram. This number will be distributed from $\phi_{min} = -\psi = -\texttt{PhiLimit}$ to $-\phi_{min} = \psi = \texttt{PhiLimit}$.

`PhiLimit` [Float] The axial acceptance angle ψ (measured in degrees).

`GenerateVolume` If set to 0 then the volume will not be generated, and 1 then it will be generated.

`GenerateSinogram` If set to 0 then the sinogram will not be generated, and 1 then it will be generated.

`NumberOfShapes` Number of primitives used.

The ordering of the parameters shown above are arbitrary. After these parameters a number of lines must follow specifying a scaling, rotation, and shifting parameters of each primitive. The line uses the following parameters in this order

`Shape`i [String] where i is 0 to `NumberOfShapes`-1.

`Type` [Integer] The type of primitive, where

- 1 A ball centered around (0,0) with radius 1 and uniform signal 1 in the ball.
- 2 A 3D Gaussian bell $\exp(-x^2 - y^2 - z^2)$.

`offx` [Float] Shift of of the primitive in the direction of the x-axis.

`offy` [Float] Shift of of the primitive in the direction of the y-axis.

`offz` [Float] Shift of of the primitive in the direction of the z-axis.

`rotx` [Float] Rotation angle of the primitive around the x-axis.

`roty` [Float] Rotation angle of the primitive around the y-axis.

`rotz` [Float] Rotation angle of the primitive around the z-axis.

`scalx` [Float] Scaling of the primitive in direction of the x-axis.

`scaly` [Float] Scaling of the primitive in direction of the y-axis.

`scalz` [Float] Scaling of the primitive in direction of the z-axis.

`power` [Float] Scaling of the value of the primitive.

The program is executed with the initialization file or `.INI`-file as sole argument. A typical `.INI`-file could be the one used to generate the Big Mickey. Note that comments can be used, such as the line above `Shape0`.

```
OutFileName=../signals/MickeyBig
XSamples=151
DeltaX=0.7
USamples=71
ThetaSamples=2
PhiSamples=2
PhiLimit=90
DeltaU=0.5
NumberOfShapes=17
GenerateVolume=1
GenerateSinogram=1
DebugLevel=Normal

         Type offx  offy   offz   rotx roty rotz scalx scaly scalz power

Shape0=  1    0.0   30.0   0.0    0    0     0    10    10    10     1
Shape1=  1  -10.7   40.5   0.0    0    0     0     5     5     5     1
Shape2=  1   10.7   40.5   0.0    0    0     0     5     5     5     1
Shape3=  1    0.0   30.0  13.0    0    0     0     3     3     3     1
Shape4=  1    0.0   24.0   9.5    0    0     0     6     2     1     1
Shape5=  1    4.5   34.5  10.0    0    0     0     2     2     2     1
Shape6=  1   -4.5   34.5  10.0    0    0     0     2     2     2     1
Shape7=  1    0.0   0.00   0.0    0    0     0    10    20    10     1
Shape8=  1  -19     20.0   0.0    0    0    30     3    15     3     1
Shape9=  1   19     20.0   0.0    0    0   -30     3    15     3     1
Shape10= 1  -18    -25.0   0.0    0    0   -30     3    15     3     1
Shape11= 1   18    -25.0   0.0    0    0    30     3    15     3     1
Shape12= 1    0.0    0.0   0.0    0    0     0     8    18     8    -1
Shape13= 1    1.0    3.0  -2.0    0    0     0     2     2     2   0.5
Shape14= 1   -2.0    2.0   2.0    0    0     0     2     2     2   0.2
Shape15= 1    0.0   30.0   0.0    0    0     0     9     9     9    -1
Shape16= 1    0.0   30.0   2.0    0    0     0     1     2     1    0.7
```

Part IV

Papers

Appendix G

Fast Radon Transform for Detection of Seismic Reflections

This appendix contains the paper *Fast Radon Transform for Detection of Seismic Reflections*, by Peter A. Toft and Kim V. Hansen [2]. The work has been carried out as a joint venture project between Ødegaard & Danneskiold-Samsøe and Department of Mathematical Modelling (before january 1. 96 Electronics Institute).

The results have been presented at the *Interdisciplinary Inversion Summer School 94* which took place in Mønsted, Denmark, and later this paper was presented at the *EUSIPCO 94* in Edinburgh, Scotland.

Fast Radon Transform for Detection of Seismic Reflections

Peter A. TOFT and Kim V. HANSEN†

*Electronics Institute, Technical University of Denmark, DK-2800 Lyngby, Denmark,
Tel/Fax +45 45 93 42 07/42 88 01 17, E-Mail: ptoft@ei.dtu.dk*
*†Ødegaard & Danneskiold-Samsøe ApS, Kroghsgade 1, DK-2100 København Ø, Denmark,
Tel/Fax +45 35 26 60 11/35 26 50 18, E-Mail: kvh@oedan.dk*

Abstract. A new method for fast detection of seimic reflections is proposed. Generally the method can be viewed as a curve matching method for images with a specific structure. In this paper the method is applied to seismic signals assembled to constitute an image in which the investigated reflections produce hyperbolic curves.
The idea of the proposed method is to estimate the reflection curves by combining the multipulse technique and the FCE-algorithm [1]. The FCE-algorithm is an algorithm used for fast curve estimation within binary images. The foundation of the FCE-algorithm is the composition of a pre-condition map which significantly reduces the computational cost compared to the traditional generalized Radon transform.
The proposed method is succesfully applied for detection of hyperbolic reflection curves within CMP-gathers.

1 Introduction

The seismic industry has developed a method of sampling seismic signals that gives multiple coverage. The recording geometry of seismic acquisition results in signals having the same mid point between source and receiver [2]. The set of signals corresponding to the same mid point is called a CMP-gather (common mid point gather).

Without any assumptions concerning the seismic signal processing, two facts of the CMP-gather are noticed. Almost the same short signal sequence (the wavelet) is observed several times within each trace and the same short signal sequence is seen along curves within several traces. These two facts form the basis of the reflection detection. The reflections represent information concerning the geological structure of the subsurface. If the subsurface is assumed to be horizontally layered it can be shown [2] that the shape of the reflection curves are hyperbolas, described by the normal moveout equation. Currently, reflections are identified from CMP-gathers by using velocity analysis succeeded by a manual picking of primary reflections. This procedure is inconvenient due to the computational burden and the required interpretation performed by an expert. These facts motivate the development of an automatic reflection identification procedure.

The key idea of the proposed method for detection of reflection curves is to combine the multipulse technique with the FCE-algorithm [1]. The multipulse technique is used to identify the wavelet in each trace, and the FCE-algorithm is used to coordinate the estimation of the reflection sequence.

The proposed method works on CMP-gathers, shot-gathers, and other types of gathers.

2 The Multipulse Technique

Recently, the multipulse excited speech coding technique, which was originally proposed by Atal *et al.* [3], has been suggested for seismic deconvolution by Cookey *et al.* [4]. Basically, the multipulse model for speech assumes the speech signal to be a result of several impulses modulated by the shape of the vocal tract filter. Within seismic deconvolution models, the input signal is often modeled as an all-pole impulse and the layered earth reflectivity model is considered as an impulse train.

Fundamentally, the reflections within each trace are estimated by minimizing the mean square error between the recorded and the modeled signal. The method is characterized by successive estimation of a pre-defined number of reflections. Thus, the multipulse technique can be used to map seismic signals in a CMP-gather into a binary image, where the image value is set to one in case of a reflection and otherwise is set to zero.

3 Fast Hyperbola Estimation

A recurring problem in computer image processing is the identification of curves with specific shapes. In digital image processing a special variant of the generalized Radon transform [5], called the Hough transform

[6, 7], is used for detection of curves in images. The Hough transform detects specific values of parameters, which characterize these curves. Spatially extended curves are transformed to produce spatially compact features in a space of possible parameter values. Thus, the generalized Radon transform converts a difficult global detection problem in the image space into a more easily solved local peak detection problem in a parameter space.

The foundation of the FCE-algorithm is the composition of a pre-condition map which reduces the computational costs of the traditional generalized Radon transform. The pre-condition map is composed of irregular regions in the parameter domain which represent the interesting areas of the parameter domain, i.e., the regions of the parameter domain which contain peaks representing curves. Initially, the FCE-algorithm estimates the pre-condition map and subsequently applies a traditional generalized Radon transform within the regions specified by the pre-condition map. To generate the pre-condition map a fast mapping procedure named *image point mapping* (*IPM*) is presented. As *IPM* maps image points into the corresponding parameter values in the parameter domain, it can profit by image points with value zero.

Assuming a horizontally layered subsurface the reflection curves within a CMP-gather are described by the normal moveout equation [2]

$$t_n = \phi(\kappa_x; t_{0,z}, \sigma_s) = \sqrt{t_{0,z}^2 + (\sigma_s \kappa_x)^2} \qquad (1)$$

where t_n denotes the two way travel time, $t_{0,z}$ the zero offset two way travel time, κ_x the offset, and σ_s the slowness. The subscripts denote corresponding discrete variables.

In this paper a linear sampling of the variables is chosen according to

$$\xi_i = \xi_{i,j} = \xi_{i,low} + j_i\, \Delta\xi_i, \quad j_i = 0, 1, \ldots, J_i - 1 \quad (2)$$

where j_i is a discrete variable, $\xi_{i,low}$ denotes the lower limit, $\Delta\xi_i$ the sampling interval and J_i is the number of samples of ξ_i.

Substitution of the linear sampling into eq. (1) gives

$$\begin{aligned} n &= \phi_d(x; z, s) \\ &= round((\phi(\kappa_x; t_{0,z}, \sigma_s) - t_{low})/\Delta t) \qquad (3) \end{aligned}$$

where $round(\cdot)$ rounds to the closest integer.

Let $d(x, n)$ denote a discrete image and let $\mathcal{D}(z, s)$ denote the discrete generalized Radon transform of $d(x, n)$ defined by

$$\mathcal{D}(z, s) = \sum_{x=0}^{X-1} d(x, \phi_d(x; z, s)) \qquad (4)$$

Normally, the use of rounding instead of e.g., linear interpolation will not cause problems. However, interpolation can easily be incorporated in eq. (3) and (4).

The discrete generalized Radon transform can be rewritten as

$$\mathcal{D}(z, s) = \sum_{x=0}^{X-1} \sum_{n=0}^{N-1} d(x, n)\, \delta(n - \phi_d(x; z, s)) \qquad (5)$$

where $\delta(\cdot)$ denotes the Kronecker delta function.

Eq. (5) shows that each image point (x, n) is transformed into a parameter curve where $\phi_d(x; z, s) = n$. Thus mapping of image points with zero value is needless, as they will not contribute to $\mathcal{D}(z, s)$.

Inversion of eq. (1) and insertion of the linear sampling relations of the variables leads to

$$\begin{aligned} z &= \Omega_d(x, n; s) \\ &= round((\sqrt{t_n^2 - (\sigma_s \kappa_x)^2} - t_{0,low})/\Delta t_0) \quad (6) \end{aligned}$$

Based on eq. (5) and (6) the estimation scheme *IPM* is defined. This can be summarized as

1. Initialize $\mathcal{D}(z, s) = 0$ for all (z, s)
2. For all image points $d(x, n)$ with a value different from zero do
   For all possible s do
   $\mathcal{D}(\Omega_d(x, n; s), s) := \mathcal{D}(\Omega_d(x, n; s), s) + d(x, n)$

In the hyperbolical case eq. (6) will describe an ellipse. This causes a problem for $|\Omega_d(x, n; s) - \Omega_d(x, n; s-1)| > 1$, which results in a perforated ellipse. This perforation problem can be eliminated simply by adding $d(n, x)$ to $\mathcal{D}(z, s)$ for the parameter values skipped between two subsequent parameter vectors. Using *IPM* the ellipses from points lying on a hyperbola might not intersect in one parameter combination, but will spread out. This problem can be eliminated using a sparse sampled parameter domain. However, *GRT* requires a dense sampling of the parameter domain to ensure that all hyperbolas are detectable.

3.1 *The FCE-algorithm*

From the characteristics of *GRT* and *IPM* it is possible to construct a fast curve parameter estimation algorithm, the *FCE-algorithm* [1], that simultaneously estimates all curve parameters within a sparse binary image.

The FCE-algorithm can be summarized as follows:

1. Design the discrete parameter domain, i.e., choose $\xi_{i,low}$, J_i and $\Delta\xi_i$, $\xi_i \in \{t_{0,z}, \sigma_s\}$.
2. Design a reduced parameter domain for *IPM* by choosing

$$\begin{aligned} J_i' &= ceil\left(\frac{J_i}{2v_{\xi_i}+1}\right) && (7) \\ \xi_{i,low}' &= \xi_{low} + v_{\xi_i}\Delta\xi_i && (8) \\ \Delta\xi_i' &= (2v_{\xi_i}+1)\Delta\xi_i && (9) \end{aligned}$$

where $ceil(\cdot)$ rounds to the nearest upper integer. Then use *IPM* to estimate $\mathcal{D}_m(z', s')$ as

$$\mathcal{D}_m(z', s') = IPM\{d(x, n)\} \tag{10}$$

3. Use a threshold to remove all insignificant parameter combinations. A suitable choice could be

$$\mathcal{D}_t(z', s') = \mu(\mathcal{D}_m(z', s') - \lambda_1\ X) \tag{11}$$

where X denotes the number of image lines, λ_1 a fraction defining the significance level, and $\mu(\cdot)$ the Hamilton step function.

4. Resample the parameter domain to obtain $\mathcal{D}_r(z, s)$. The resampling process can be written as

$$\mathcal{D}_r(z, s) = \mathcal{D}_t(z', s') \tag{12}$$

$$j'_i = round\left(\frac{j_i - v_{\xi_i}}{2v_{\xi_i} + 1}\right) \tag{13}$$

This gives a full size parameter domain where the interesting regions have value one while all other regions have value zero.

5. Use *GRT* to estimate $\mathcal{D}(z, s)$ in the regions of the parameter domain where $\mathcal{D}_r(z, s)$ is not equal to zero.

$$\mathcal{D}(z, s) = GRT\{d(x, n) |\ \mathcal{D}_r(z, s) \neq 0\} \tag{14}$$

6. As in step three use the threshold function with significance level λ_2. After the thresholding all points in the parameter domain having a value different from zero represent curve parameters.

The significance levels λ_1 and λ_2 must be chosen in accordance with the properties of the image. In general λ_1 and λ_2 are not critical. A suitable choice of λ_1 and λ_2 could be $0.1 - 0.7$ and $0.4 - 0.8$, respectively. There are two reasons for a low acceptance level in step three. Firstly, *IPM* has a tendency to blur in the parameter domain and secondly, only regions of the parameter domain ensured not to contain image curves must be left out. A higher significance level in step six (λ_2) is justified by the required certainty for correct curve identification.

4 Numerical Example

To illustrate the performance of the presented method for determination of reflection curves within CMP-gathers a numerical example is given.

Consider a CMP-gather where the image values are represented by continuous values. The CMP-gather can be mapped into a binary image $d(x, n)$ by deconvolution based on the multipulse technique [4, 8]. Figure 1.left shows a synthetic CMP-gather and Figure 1.right the thresholded image obtained by multipulse technique deconvolution. The CMP-gather is composed of seven reflection curves. Concerning the deconvolution fifteen reflections are chosen to be estimated within each trace, which implies that eight erroneous reflections within each trace have been found. These eight reflection will act as uncorrelated noise.

Next, Figure 2 shows the parameter domain obtained by use of *IPM*, where $v_{t_0} = 2$ and $v_\sigma = 4$. Figure 3 shows the parameter domain achieved by applying *GRT* in the parameter domain region specified by the thresholded and resampled parameter domain obtained by *IPM*. Within the thresholding process a significance level of $\lambda_1 = 0.5$ has been used. Finally, Figure 4 shows the identified reflection hyperbolas, where a significance level of $\lambda_2 = 0.8$ has been used in the thresholding. As seen all reflection curves have been found.

5 Conclusion

A new method for fast detection of seismic reflections is presented. Generally, the method can be viewed as a curve matching method for images with a specific structure.

The foundation of the proposed method is to estimate the reflection curves by combining the multipulse technique [3, 4] and the FCE-algorithm [1]. The multipulse technique is used for wavelet identification within each trace, and the FCE-algorithm is used to coordinate the wavelet identification between the individual traces.

The proposed method is succesfully applied for detection of hyperbolic reflection curves within CMP-gathers.

References

[1] K. V. Hansen and P. A. Toft, "Fast Curve Estimation Using Pre-Conditioned Generalized Radon Transform," Submitted for publication, August 1993.

[2] W. A. Schneider, "The Common Depth Point Stack," *Proc. IEEE*, vol. 72, no. 10, pp. 1238–1254, Oct. 1984.

[3] B. S. Atal and J. R. Remde, "A New Model of LPC Excitation for Producing Natural-Sounding Speech at Low Bit Rates," *IEEE ICASSP*, vol. 1, 1982, pp. 614–617.

[4] M. Cookey, H. J. Trussel and I. J. Won, "Seismic Deconvolution by Multipulse Methods," *IEEE Trans. ASSP*, no. 1, vol. 38, pp. 156–160, Jan. 1990.

[5] J. Radon, “Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten,” *Ber. Ver. Sächs. Akad. Wiss. Leipzig, Math-Phys. Kl.*, vol. 69, pp. 262–277, April 1917.

[6] R. O. Duda and P. E. Hart, “Use of the Hough Transformation to Detect Lines and Curves in Pictures,” *Com. ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1972.

[7] J. Illingworth and J. Kittler, “A Survey of the Hough Transform,” *Computer Vision, Graphics, and Image Processing*, vol. 44, pp. 87–116, 1988.

[8] K. V. Hansen and J. Larsen, “An Algorithm for Successive Identification of Reflections,” *IEEE Trans. IP*, vol. 3, no. 3, May 1994.

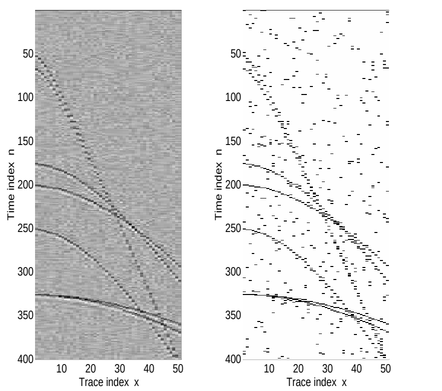

Figure 1: Left: CMP-gather. Right: Thresholded CMP-gather.

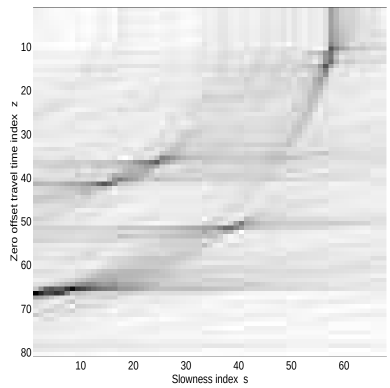

Figure 2: IPM-domain.

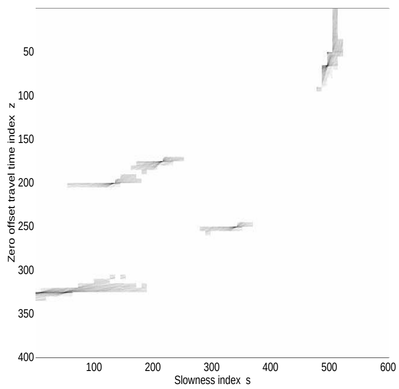

Figure 3: GRT-domain.

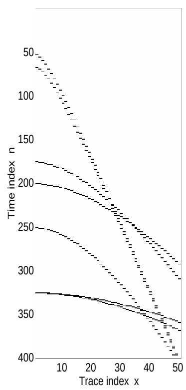

Figure 4: Identified reflection curves.

Appendix H

Fast Curve Estimation Using Pre-Conditioned Generalized Radon Transform

This appendix contains a pre-print of the paper *Fast Curve Estimation Using Pre-Conditioned Generalized Radon Transform*, by Kim V. Hansen and Peter A. Toft [1]. The work has been carried out as a joint venture project between Ødegaard & Danneskiold-Samsøe and Department of Mathematical Modelling (before january 1. 96 Electronics Institute).

The paper has been accepted for publication in the IEEE Transactions on Image Processing and should appear in the December 96 issue.

Fast Curve Estimation Using Pre-Conditioned Generalized Radon Transform

Kim V. Hansen, Oticon, Strandvejen 58, 2900 Hellerup, Phone +45 39177120,
e-mail kvh@Oticon.Oticon.dk400.dk
Peter A. Toft, Department of Mathematical Modelling, Building 349,
Technical University of Denmark, 2800 Lyngby, Denmark, Phone +45 45934207,
e-mail ptoft@ei.dtu.dk

Abstract

A new algorithm for fast curve parameter estimation based on the generalized Radon transform is proposed. The algorithm works on binary images, obtained, e.g., by edge filtering or deconvolution. The fundamental idea of the suggested algorithm is the use of a pre-condition map to reduce the computational cost of the generalized Radon transform. The pre-condition map is composed of irregular regions in the parameter domain, which contain peaks that represent curves in the image. To generate the pre-condition map, a fast mapping procedure named *image point mapping* is developed. As the image point mapping scheme maps image points into the corresponding parameter values in the parameter domain, it is possible to improve computational efficiency be recognizing image points with value zero. Initially, the suggested algorithm estimates the pre-condition map and subsequently applies the generalized Radon transform within the regions specified by the pre-condition map. The required parameter domain sampling and the resulting blurring are also investigated.

The suggested algorithm is successfully applied to the identification of hyperbolas in seismic images, and two numerical examples are given.

I Introduction

A recurring problem in computer image processing is the identification of curves with specific shapes. The transform developed by Hough [1] for detection of complex patterns in binary $\{0,1\}$ digital images has been discussed by several authors, e.g., [2], [3] and generalized for multi-dimensional pattern detection in [4]. The Hough transform can be deduced [5] as a special case of a more general transform, known as the Radon transform [6]. The Radon transform is a mapping from an image domain to a parameter domain, where the parameters characterize the curves to be identified. The Radon transform can be generalized to handle arbitrary curves [7]. The Hough version of the generalized Radon transform detects specific values of parameters, as spatially extended curves are transformed to produce spatially compact features in the parameter domain. In this way, the generalized Radon transform converts a difficult global detection problem in the image domain into a more easily solved local peak detection problem in the parameter domain.

In recent years progress has been made to understand and increase the speed of the generalized Radon transform [8, 9, 4]. Investigations concerning both the traditional and more recent generalized Radon transform for curve identification schemes point out two problems with respect to computational cost. First, the estimation schemes are not capable of exploiting image points with zero value (zero image points), and second, estimation is computationally expensive, as it estimates unneeded information.

This is a consequence of parameter estimation, in areas of the parameter domain where curve parameters are very unlikely.

The basis for curve parameter estimation as described in this paper is a binary image. The binary image can be produced in several ways, e.g., by edge filtering or deconvolution.

In this paper, a new algorithm for fast curve estimation, called the FCE-algorithm, is presented. The key idea of the FCE-algorithm is to pre-condition the parameter domain, creating irregular regions corresponding to the parameter regions of interest. Furthermore, the FCE-algorithm takes advantage of zero image values in generating the pre-condition map. Initially, the FCE-algorithm identifies regions of the parameter domain which contain peaks representing curves in the image. Subsequently, it estimates a traditional generalized Radon transform within these regions.

The identification of hyperbolas is of particular interest within seismic signal processing. Over the years the seismic industry has developed a method of recording seismic signals resulting in images in which the investigated reflections produce curves. The recording geometry of seismic data acquisition results in signals having the same geometrical mid point between source and receiver, see, e.g., [10]. The set of signals corresponding to the same mid point is called a common mid point gather (CMP-gather). A common technique in seismic data analysis for estimation of hyperbola parameters is velocity analysis, e.g., [11]. Within the presented work some features from velocity analysis are incorporated, however, the presented method has less computational cost.

Recently, the multipulse excited speech coding technique, which was originally proposed by Atal *et al* [12], has been suggested for seismic deconvolution by Cookey *et al.* [13]. Basically, the multipulse model for speech assumes the speech signal to be a result of several impulses transformed by the shape of the vocal tract filter. In seismic deconvolution, the input signal is often modeled as an all-pole impulse and the layered earth reflectivity model is considered as an impulse train, see, e.g., [14]. Thus, the multipulse technique can be used to map seismic signals in a CMP-gather into a binary image, where the image value is set to one in case of a reflection and zero otherwise. The reflections represent information concerning the geological structure of the subsurface.

The traditional generalized Radon transform is described in section II. Creation of the preconditioning map is described in section III, where the *image point mapping* procedure is developed. In section IV, parameter domain sampling is discussed and section V describes parameter domain blurring, which is a result of discretization. The FCE-algorithm is presented in section VI, and is applied to two numerical examples for detection of hyperbolas in section VII. Regarding notation, scalars are denoted by letters, and vectors by bold-faced letters.

II The Generalized Radon Transform

Let $f(\kappa, t)$ be a continuous signal of the continuous variables κ and t and let $\boldsymbol{\xi}$ denote an η-dimensional parameter vector defined as

$$\boldsymbol{\xi} = (\xi_1, \ldots, \xi_i, \ldots, \xi_\eta) \qquad (1)$$

where (κ, t) is called the image domain, $\boldsymbol{\xi}$ the parameter domain, and $f(\kappa, t)$ the image.

A. *The Continuous Generalized Radon Transform*

Let $F(\boldsymbol{\xi})$ denote the continuous generalized Radon transform of $f(\kappa, t)$, defined as

$$F(\boldsymbol{\xi}) = \int_{-\infty}^{\infty} f(\kappa, \phi(\kappa; \boldsymbol{\xi}))\, d\kappa \qquad (2)$$

where $t = \phi(\kappa; \boldsymbol{\xi})$ denotes the transformation curve. The generalized Radon transform is the integration of $f(\kappa, t)$ along the transformation curve.

Other definitions of the generalized Radon transform can be found in the literature [5], [6]. Equation 2 can be interpreted as a generalization of the Slant Stacking technique used in geophysics, e.g., [7], but the new ideas presented in this paper can be modified to confirm with this literature.

B. *The Discrete Generalized Radon Transform*

Let $\boldsymbol{j}$ denote the η-dimensional discrete index parameter vector defined as

$$\boldsymbol{j} = (j_1, \ldots, j_i, \ldots, j_\eta) \tag{3}$$

The correspondence between the index vector $\boldsymbol{j}$ and the sampled version of the parameter vector $\boldsymbol{\xi}$ can be written as

$$\boldsymbol{\xi} = \boldsymbol{\xi}_{\boldsymbol{j}} = \boldsymbol{\theta}(\boldsymbol{j}), \quad \text{where} \quad \xi_i = \theta_i(j_i) \tag{4}$$

where $\theta_i(j_i)$ is a parameter sampling function.

If a uniform sampling of the parameter domain is chosen, the function θ_i can be written

$$\xi_i = \theta_i(j_i) = \xi_{i,min} + j_i \; \Delta\xi_i, \quad j_i = 0, \ldots, J_i - 1 \tag{5}$$

where $\xi_{i,min}$ denotes the lower limit and $\Delta\xi_i$ the sampling interval of ξ_i.

The parameter domain and image domain sampling are assumed to be uniform. The correspondence between the discrete image domain indices (m, n) and the continuous variables (κ, t) can be written as

$$\kappa = \kappa_m = \kappa_{min} + m \; \Delta\kappa, \; m = 0, 1, \ldots, M - 1 \tag{6}$$

$$t = t_n = t_{min} + n \; \Delta t, \; n = 0, 1, \ldots, N - 1 \tag{7}$$

where κ_{min} and t_{min} denote the lower limits and $\Delta\kappa$ and Δt the sampling interval of κ and t, respectively.

Substitution of Equation 4, 6 and 7 into the transformation curve gives

$$t = t_{min} + n \; \Delta t = \phi(\kappa_{min} + m \; \Delta\kappa; \boldsymbol{\theta}(\boldsymbol{j})) \tag{8}$$

From to Equation 8, the discrete index transformation curve $\phi_d(m; \boldsymbol{j})$ can be expressed as

$$\begin{aligned} \phi_d(m; \boldsymbol{j}) &= n \\ &= \operatorname{round}\left(\frac{\phi(\kappa_{min} + m \; \Delta\kappa; \boldsymbol{\theta}(\boldsymbol{j})) - t_{min}}{\Delta t}\right) \end{aligned} \tag{9}$$

where round$(\cdot)$ rounds to the closest integer.

Let $d(m, n)$ denote the discretized version of the signal $f(\kappa, t)$, i.e., $d(m, n) = f(\kappa_m, t_n)$ and let $D(\boldsymbol{j})$ denote the discrete generalized Radon transform (GRT) of $d(m, n)$ defined as

$$D(\boldsymbol{j}) = \sum_{m=0}^{M-1} d(m, \phi_d(m; \boldsymbol{j})) \tag{10}$$

Notice, that the term $d\kappa$ has been omitted in the discrete generalized Radon transform as it is a constant due to the uniform sampling of κ. The use of rounding instead of, e.g., linear interpolation will normally not lead to problems. If necessary, interpolation can easily be incorporated in Equations 9 and 10.

III Image Point Mapping

From Equation 2, the continuous generalized Radon transform $F(\boldsymbol{\xi})$ can be written as

$$F(\boldsymbol{\xi}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\kappa, t) \, \delta(t - \phi(\kappa; \boldsymbol{\xi})) \, dt \; d\kappa \tag{11}$$

where $\delta(\cdot)$ is the Dirac delta function.

Analogously, the discrete generalized Radon transform, from Equation 10, can be written as

$$D(\boldsymbol{j}) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} d(m, n) \, \delta(n - \phi_d(m; \boldsymbol{j})) \tag{12}$$

where $\delta(\cdot)$ denotes the Kronecker delta function.

Equation 12 shows that each image point (m, n) is transformed into a parameter curve where $\phi_d(m; \boldsymbol{j}) = n$. Thus mapping of image points with zero value is needless as they will not contribute to $D(\boldsymbol{j})$. The proposed estimation scheme for the Radon transform accounts for the fact that zero image values do not contribute, and does not include them in the summation in Equation 12. This has the potential to reduce the computational cost considerably.

Now, assume that ϕ is invertible in one of the parameters ξ_i, e.g., ξ_η. The inverse function $\phi_{\xi_\eta}^{-1}$ with respect to ξ_η can be written as

$$\xi_\eta = \phi_{\xi_\eta}^{-1}(\kappa, t; \boldsymbol{\xi}_r), \quad \boldsymbol{\xi}_r = (\xi_1, \ldots, \xi_{\eta-1}) \tag{13}$$

Furthermore, assume that the sampling function θ_η is invertible. Then

$$\begin{aligned} j_\eta &= \text{round}(\theta_\eta^{-1}(\xi_\eta)) \\ &= \text{round}(\theta_\eta^{-1}(\phi_{\xi_\eta}^{-1}(\kappa, t; \boldsymbol{\xi}_r)) \\ &\equiv \Omega_d(m, n; \boldsymbol{j}_r),\ \boldsymbol{j}_r = (j_1, \ldots, j_{\eta-1}) \end{aligned} \quad (14)$$

Equations 12 and 14, are the basis for an estimation scheme for the generalized Radon transform which is referred to here as *Image Point Mapping* (IPM). The key steps of IPM can be summarized as

1. Initialize $D(\boldsymbol{j}) = 0$ for all $\boldsymbol{j}$
2. For all image points $d(m, n)$ with a value different from zero do
   For all possible $\boldsymbol{j}_r$ do
   $\boldsymbol{j} = (\boldsymbol{j}_r, \Omega_d(m, n; \boldsymbol{j}_r))$
   $D(\boldsymbol{j}) := D(\boldsymbol{j}) + d(m, n)$

Notice, that IPM is a generalization of the Hough transform [1] extended to handle arbitrary transformation curves. Like the Hough transform, the objective of IPM is parameter identification.

The fundamental difference between GRT and IPM is, that GRT requires rounding in the image domain, whereas IPM rounds in the parameter domain, as schematically shown in Figure 1.

The computational complexities of GRT and IPM, respectively, are

$$\begin{aligned} W_{GRT} &= O\left(M \prod_{i=1}^{\eta} J_i\right) \\ W_{IPM} &= O\left((M\,N)_r \prod_{i=1}^{\eta-1} J_i\right) \end{aligned} \quad (15)$$

where $(M\,N)_r$ indicates that only a reduced number of image points (nonzero values) are to be transformed. In Equation 15, the cost to test for nonzero values is assumed to be negligible.

The major advantage of IPM over GRT is the ability of IPM to ignore image points with a value of zero, making IPM especially well suited for sparse binary images.

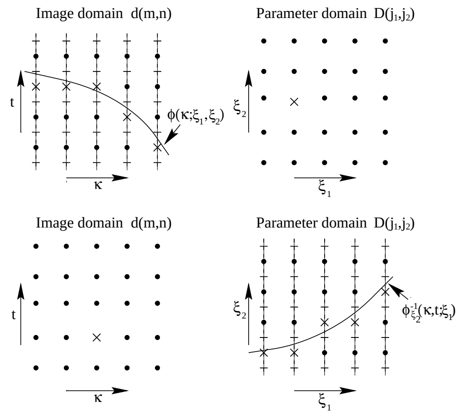

Figure 1: Top: Rounding by GRT in the image domain corresponding to a point in the parameter domain. Bottom: Rounding by IPM in the parameter domain corresponding to a point in the image domain.

If the absolute value between two successive j_η values is greater than one, i.e.,

$$|\Omega_d(m, n; \boldsymbol{j}_r + \boldsymbol{\varepsilon}) - \Omega_d(m, n; \boldsymbol{j}_r)| > 1 \quad (16)$$

where vector $\boldsymbol{\varepsilon}$ contains zeros at all $\eta - 1$ entries and holds unit value at entry i, i.e., $\varepsilon_i = 1$, it is seen that IPM will map the image point (m, n) into a perforated hypercurve, i.e., a hypercurve containing missing sections. Identification of curve parameters can be difficult due to the presence of perforation holes depending on the shape and depth of the holes.

For $\eta = 2$ the perforation problem can be eliminated simply by adding $d(m, n)$ to $D(\boldsymbol{j})$ for the parameter values skipped between two successive parameter vectors. Consider, for example, the hyperbolic transformation curve as illustrated in Figure 2, where the parameter domain is described by $(j_1, j_2) = (s, z)$.

Figure 2 shows the mapping of an image point (m^*, n^*) to the parameter domain, that is, discretized parameter point accumulators to which the value $d(m^*, n^*)$ of the image point must be added. To investigate the perforation problem, two parameter points $(s^* - 1, z^* + 3)$ and (s^*, z^*) on the inverse transformation curve are

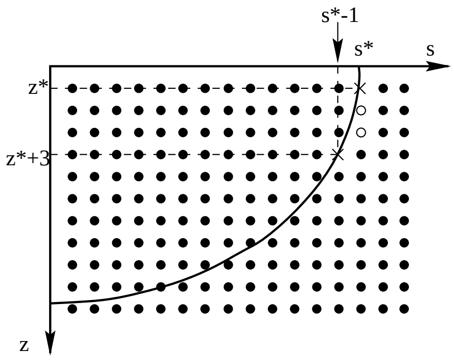

Figure 2: Illustration of the perforation problem.

considered. Following the inverse transformation curve from (s^*-1, z^*+3) to (s^*, z^*) along the s axis gives only the two parameter points (s^*-1, z^*+3) and (s^*, z^*). However, following the inverse transformation curve along the z axis also gives the two intervening points (s^*, z^*+2) and (s^*, z^*+1). One way of handling such intervening points is to add the value of the image point to the parameter points skipped between two successive parameter points on the inverse transformation curve. Therefore, in this example, $d(m^*, n^*)$ must be added to the skipped parameter points (s^*, z^*+2) and (s^*, z^*+1) when following the s axis.

IV Parameter Domain Sampling

For a given image $d(m,n)$ an exact determination of $\Delta\xi_i$, $\xi_{i,min}$ and J_i, which match the image, is, in general, not possible. However, some guidelines can be stated. First, some of the parameters will be bounded by the underlying physics, e.g., $\xi_{i,min}$ and $\xi_{i,max}$ will normally be limited. Second, the parameter domain can be limited by requiring that at least a fraction β of the image points addressed by the transformation curve $\phi_d(m;\boldsymbol{j})$ lie inside the image, i.e.,

$$\sum_{m=0}^{M-1} I\left\{t_{min} < \phi(\kappa_m;\boldsymbol{\xi}) < t_{max}\right\} > \beta\, M,\ \forall \boldsymbol{\xi} \tag{17}$$

where $0 < \beta < 1$, and $I\{\cdot\}$ equals 1 when the logical expression is true and 0 otherwise.

Concerning use of the GRT, the sampling intervals $\Delta\xi_i$ can be chosen by requiring that

$$\max\{|\phi(\kappa;\boldsymbol{\xi}+\boldsymbol{\gamma_i}) - \phi(\kappa;\boldsymbol{\xi})|\} \;=\; \Delta t\ \ \forall\, \kappa, \boldsymbol{\xi}, \boldsymbol{\gamma_i} \tag{18}$$

where $\boldsymbol{\gamma_i}$ is a η-dimensional vector containing zeros at all entries except entry i which is $\Delta\xi_i$.

Equation 18 states that two adjacent $\boldsymbol{\xi}$-vectors give $t = \phi(\kappa;\boldsymbol{\xi})$ values which cannot be separated by more than one sample in the t-direction. This design criterion is not optimal as it leads to an unnecessarily dense sampling of certain parts of the parameter domain, however, it can be used as an upper sampling rate limit. Increasing the sampling rate of the parameter domain above a certain limit will not improve the resolution obtained by GRT as adjacent $\boldsymbol{j}$ vectors will result in the same curve $n = \phi_d(m;\boldsymbol{j})$. For GRT, a coarse sampling cannot guarantee a given curve will be detected, due to the fact that no transformation curve $\phi_d(m;\boldsymbol{j})$ can be guaranteed to follow the image curve perfectly.

Use of IPM with a dense parameter domain sampling according to Equation 18 produces curves $j_\eta = \Omega_d(m,n;\boldsymbol{j}_r)$ for (m,n) values corresponding to an image curve which do not intersect in a parameter point $\boldsymbol{j}$, but will spread out over several parameter points. The reason for this spread is the individual treatment of image points as regions of zero size and not regions of, e.g., rectangular shape. This spread can be compensated by the use of a coarse sampling of the parameter domain. In addition, a coarse sampling will lead to a lower computational cost.

Summarizing, GRT requires a dense parameter domain sampling, while IPM gives rise to blurring in the parameter domain in the case of dense parameter domain sampling, but works well with a coarse sampling of the parameter domain. This observation is the basis for the curve parameter estimation algorithm, presented in section VI.

V Parameter Domain Blurring

Parameter domain blurring achieved by use of GRT can be interpreted in a way that makes it usable for parameter clustering. According to Equation 2 the continuous transformation curve can be written as

$$t = \phi(\kappa; \boldsymbol{\xi}), \qquad \kappa \in [\kappa_{min}, \kappa_{max}] \tag{19}$$

Let Δt denote the sampling interval of t and let $\alpha\Delta t$ be a given uncertainty of t, e.g., $\alpha = 0.5$. For a parameter vector $\boldsymbol{\xi} = \boldsymbol{\theta}(\boldsymbol{j})$ to lie inside a band of width $2\alpha\Delta t$, symmetrically positioned around the image curve $t = \phi(\kappa; \boldsymbol{\xi}')$, it must satisfy the following inequality

$$t - \alpha\ \Delta t < \phi(\kappa; \boldsymbol{\xi}) < t + \alpha\ \Delta t \tag{20}$$

$$t = \phi(\kappa; \boldsymbol{\xi}'), \quad \forall \kappa \in [\kappa_{min}, \kappa_{max}] \tag{21}$$

Several parameter vectors $\boldsymbol{\xi}$ will correspond to image points within the band specified in Equation 21. Therefore, in general, a band in the image around a given curve described by the parameter vector $\boldsymbol{\xi}'$ will give rise to a region in the parameter space which consists of parameter vectors that correspond to curves inside the image band. These regions are named *clusters* and the corresponding image curve with parameter $\boldsymbol{\xi}'$ is called a *center curve*. An example of a curve band around a center curve and the corresponding parameter cluster is illustrated in Figure 3.

Unfortunately, the center curves are normally unknown, and it is impossible to determine the clusters. However, it is possible to determine whether two parameter sets $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ belong to the same cluster. If two parameter vectors belong to the same cluster they must satisfy

$$|\phi(\kappa; \boldsymbol{\xi}_1) - \phi(\kappa; \boldsymbol{\xi}_2)| < 2\ \alpha\ \Delta t, \ \ \kappa \in [\kappa_{min}, \kappa_{max}] \tag{22}$$

Parameter domain points may be gathered into regions or clusters, with the guarantee that

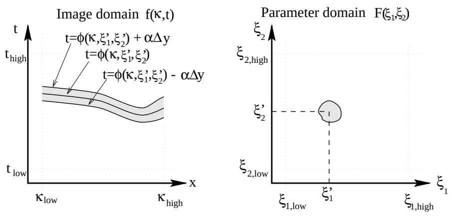

Figure 3: Left: Center curve with an uncertainty of $2\alpha\Delta t$ indicated. Right: Cluster corresponding to center curve.

all possible image curves will be represented by only one cluster in the parameter domain. This partitions the parameter domain into irregular regions, which reflect the information level of the image. Clustering can also be used to estimate parameter uncertainties, e.g., the maximum and minimum parameter value within the cluster can be used to give an estimate of the parameter uncertainty as, e.g., $\frac{1}{2}(\xi_{i,max} - \xi_{i,min} + \Delta\xi_i)$. Parameter domain clustering can be performed either before or after the transform, depending on the purpose.

VI The Fast Curve Estimation Algorithm

Consider an image where the image values are represented by continuous values. This image can be mapped into a binary image $d(m, n)$, e.g., by edge filtering [15], [16] or by deconvolution [17], [13]. The proposed algorithm estimates the parameters of curves in the binary image $d(m, n)$, e.g., lines or hyperbolas.

It is well-known that direct use of the GRT is computationally expensive, and gives a lot of unnecessary information concerning estimation of curve parameters. In the light of the characteristics of GRT and IPM, a fast curve parameter estimation algorithm, the *FCE-algorithm*, is proposed which simultaneously estimates all parameters of curves having a specific shape, e.g., lines or hyperbolas. The FCE-algorithm uses IPM as a pre-conditioning procedure for GRT by selecting the regions of interest in the parameter domain. IPM is suitable for a rapid determination of regions of interest, as it works well on a coarsely sampled parameter domain and is capable of ignoring image points with a value of zero. Another important circumstance is that GRT relates the uncertainty on the estimated curve parameters to the image domain sampling, while IPM relates it to the parameter domain sampling. The FCE-algorithm is shown in Figure 4, and is summarized as follows:

1. Design the discrete parameter domain, i.e., choose $\xi_{i,min}$, J_i and $\Delta\xi_i$.

2. Design a reduced parameter domain for IPM by choosing

$$J_i' = \text{ceil}\left(\frac{J_i}{2v_{\xi_i}+1}\right) \quad (23)$$

$$\xi_{i,min}' = \xi_{min} + v_{\xi_i}\Delta\xi_i \quad (24)$$

$$\Delta\xi_i{}' = (2v_{\xi_i}+1)\Delta\xi_i \quad (25)$$

   where ceil$(\cdot)$ rounds to the nearest upper integer, and v_{ξ_i} is an integer related to the resampling (see below). Then use IPM to estimate $D_{ipm}(\boldsymbol{j'})$ as

$$D_{ipm}(\boldsymbol{j'}) = \text{IPM}\{d(m,n)\} \quad (26)$$

3. Use the threshold function T to give

$$D_t(\boldsymbol{j'}) = T(D_{ipm}(\boldsymbol{j'})) \quad (27)$$

   The threshold function T is designed to remove all insignificant parameter combinations and a suitable choice might be

$$T(D_{ipm}(\boldsymbol{j'})) = \mu(D_{ipm}(\boldsymbol{j'}) - \lambda_1 M) \quad (28)$$

   where M denotes the number of image lines, λ_1 a fraction defining the significance level, and $\mu(\cdot)$ the Hamilton step function. Having an image with one curve, IPM gives a corresponding parameter domain value of M. Choosing a threshold level of $\lambda_1 M$ allows some deviation for the image curve from the integration curve, e.g., to handle holes.

4. Resample the parameter domain to obtain $D_r(\boldsymbol{j})$. The resampling process can be written as

$$D_r(\boldsymbol{j}) = D_t(\boldsymbol{j'}) \quad (29)$$

$$j_i' = \text{round}\left(\frac{j_i - v_{\xi_i}}{2v_{\xi_i}+1}\right) \quad (30)$$

   This quickly fills the entire parameter domain where the interesting regions have value one while other regions have value zero.

5. Use GRT to estimate $D(\boldsymbol{j})$ in the regions of the parameter domain where $D_r(\boldsymbol{j})$ is not equal to zero.

$$D(\boldsymbol{j}) = \text{GRT}\{d(m,n)\,|D_r(\boldsymbol{j}) \neq 0\} \quad (31)$$

6. As in step three, use the threshold function with a new significance level λ_2 to account for use of the GRT. After the thresholding, all points in the parameter domain having a value different from zero represent curve parameters.

7. Finally, the identified parameters are clustered into common image curves.

The resampling process takes each parameter point corresponding to the sampling interval $\boldsymbol{\Delta\xi'}$ and extends it to several parameter points corresponding to the sampling interval $\boldsymbol{\Delta\xi}$, $\Delta\xi_i < \Delta\xi_i'$. Thus, the FCE-algorithm operates initially in a coarsely sampled parameter domain, using IPM for determination of regions of interest. Subsequently, the sampling is refined to the required level and GRT is applied within the regions of interest.

The significance levels λ_1 and λ_2 must be chosen to reflect the parameter domain values, that can be accepted as peaks corresponding to curves in the image. In general, the precise values of λ_1 and λ_2 are not critical. For the examples, suitable ranges of values for λ_1 and λ_2 are $0.1 - 0.7$ and $0.4 - 0.8$, respectively. There are two reasons for the use of a lower acceptance level in step three. First, IPM has a tendency to blur in the parameter domain, and second,

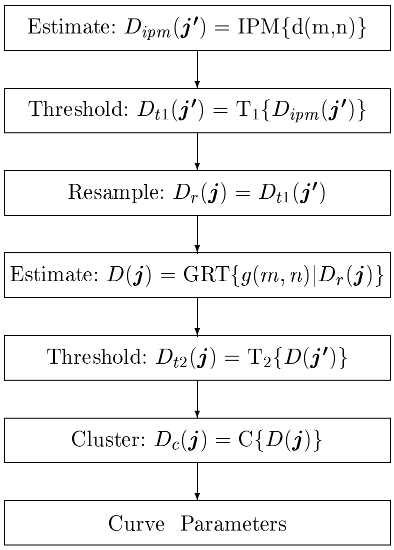

Figure 4: Flow diagram of the FCE-algorithm.

only regions of the parameter domain ensured not to contain image curves must be removed. A higher significance level in step six (λ_2) is justified by the required certainty for correct curve identification.

VII The Hyperbolic Transformation Curve

Consider a seismic CMP-gather. The multipulse technique can be used to map the seismic signals in the CMP-gather into a binary image $d(m,n)$, where $d(m,n)$ is set to one if position n of trace number m contains a reflection, and to zero otherwise. If the subsurface is assumed to be horizontally layered, it can be shown [10] that the shape of the reflection curves are hyperbolas. Traditionally, the reflection curves within the CMP-gather are described by the *normal moveout equation* [10], which can be written as

$$\phi(\kappa;\sigma,t_0) = t = \sqrt{t_0^2 + (\sigma\ \kappa)^2},\ \kappa \in [\kappa_{min}, \kappa_{max}] \tag{32}$$

where t denotes the two way travel time with distance κ between source and receiver, t_0 the zero offset two way travel time, and finally σ denotes the slowness. The variable κ is also known as the offset.

According to Equation 32, $\phi(\kappa;\sigma,t_0)$ can easily be inverted with respect to t_0 to yield

$$t_0 = \phi_{t_0}^{-1}(\kappa,t;\sigma) = \sqrt{t^2 - (\sigma\ \kappa)^2} \tag{33}$$

Both t_0 and σ are uniformly sampled and the parameter domain sampling is chosen in agreement with Equation 18. The discretized parameter domain is denoted z and s corresponding to t_0 and σ, respectively,

$$t_0 = t_{0,low} + z\ \Delta t_0,\ z = 0,\ldots,Z-1 \tag{34}$$
$$\sigma = \sigma_{low} + s\ \Delta\sigma,\ s = 0,\ldots,S-1 \tag{35}$$

The curve perforation described in section III occurs in the present case of a hyperbolic transformation curve. It is, however, eliminated by adding $d(m,n)$ to $D(s,z)$ for the parameter values skipped between two successive parameter vectors.

Assume the parameter set $(\sigma_0, t_{0,0})$ corresponds to a center curve. The parameter sets (σ, t_0) corresponding to image curves within the band of width $2\alpha\Delta t$, symmetrically positioned around the image curve $t = \phi(\kappa;\sigma_0,t_{0,0})$, can according to Equation 21 be written as

$$t - \alpha\ \Delta t < \sqrt{t_0^2 + (\sigma\ \kappa)^2} < t + \alpha\ \Delta t \tag{36}$$
$$t = \sqrt{t_{0,0}^2 + (\sigma_0\ \kappa)^2}\ \forall\kappa \in [\kappa_{min}, \kappa_{max}] \tag{37}$$

Figure 5 illustrates four clusters in the parameter domain. The four clusters correspond to four center curves with an uncertainty of $10\Delta t$, i.e., $\alpha = 5$. Formulas for the cluster shape is shown in appendix A.

To illustrate the performance of the FCE-algorithm two numerical examples will be given.

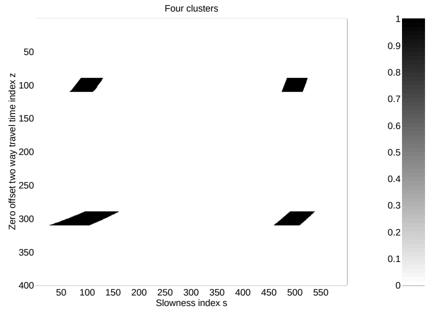

Figure 5: Four clusters in the parameter domain corresponding to an uncertainty of $10\Delta t$, i.e., $\alpha = 5$.

A. *Example 1*

In this example, the image is composed of eight hyperbolas, assembled into four groups of two hyperbolas each, as shown in Figure 6. Using GRT according to the settings in Table 1 leads to eight peaks in the parameter domain as shown in Figure 7. Thus, although the eight curves intersect and are very close, GRT is able to separate the curves in the parameter domain into eight separate peaks.

Image domain		Parameter domain	
Par.	Value	Par.	Value
M	50	S	300
N	400	Z	400
$\Delta\kappa$	$50\ m$	$\Delta\sigma$	$1.569 \cdot 10^{-6}\ s/m$
Δt	$4 \cdot 10^{-3} s$	Δt_0	$4 \cdot 10^{-3} s$
κ_{min}	$100\ m$	σ_{min}	$2.3 \cdot 10^{-4}\ s/m$
t_{min}	$0\ s$	$t_{0,min}$	$0s$
α	1.5	λ_1	0.5
		λ_2	0.75
		v_σ	4
		v_{t_0}	2

Table 1: Image and parameter domain settings.

The FCE-algorithm is used for fast detection of the interesting regions in parameter domain resulting from GRT. The following five figures show each step of the FCE-algorithm. Figure 8 shows the coarsely sampled parameter domain

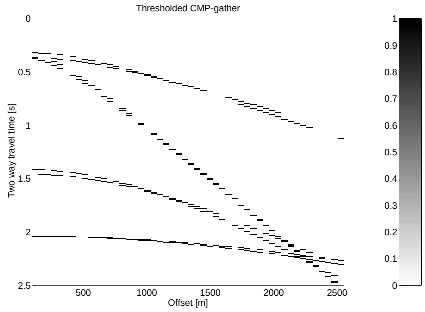

Figure 6: Synthetic binary CMP-gather composed of eight hyperbolas.

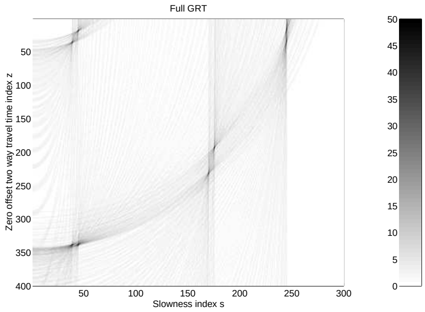

Figure 7: Full GRT estimated parameter domain.

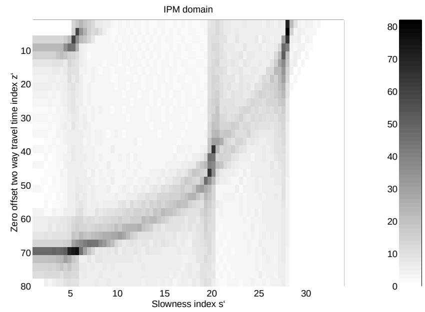

Figure 8: Parameter domain estimated by use of IPM.

obtained using IPM with $v_\sigma = 4$ and $v_{t_0} = 2$.

The parameter domain is thresholded using a significance level $\lambda_1 = 0.5$, and the resulting binary parameter domain is shown in Figure 9.

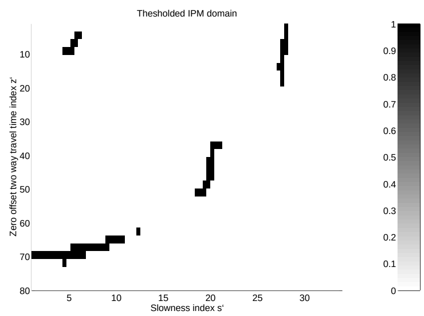

Figure 9: Thresholded IPM estimated parameter domain, where $\lambda_1 = 0.5$.

The binary parameter domain is then resampled to a full size parameter domain as shown in Figure 10.

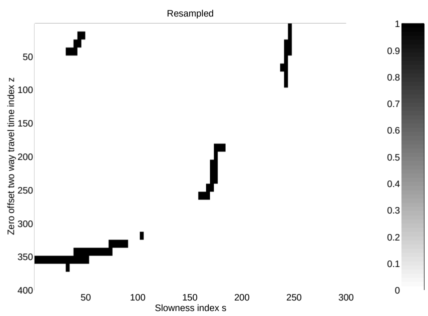

Figure 10: Resampled parameter domain.

Comparing this pre-condition map with Figure 7, indicates that all regions containing curves are contained within the pre-condition map. Next, Figure 11 shows the parameter domain obtained by using GRT within the regions specified by the pre-condition map.

Finally, the pre-conditioned GRT parameter domain is thresholded using a threshold level $\lambda_2 = 0.75$, and the result is shown in Figure 12.

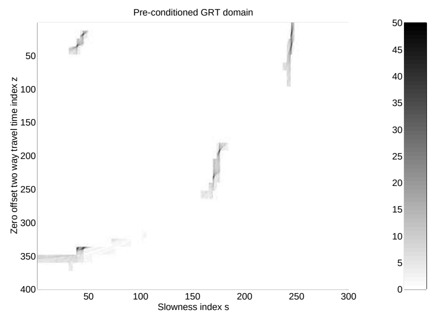

Figure 11: Parameter domain obtained using GRT within the regions specified by the resampled parameter domain shown in Figure 10.

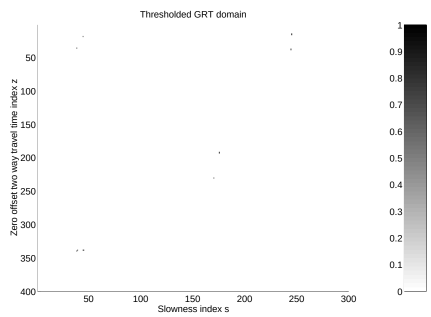

Figure 12: Thresholded GRT estimated parameter domain using $\lambda_2 = 0.75$.

The estimated parameter vectors are given in Table 2, where the clustering process has used a value of $\alpha = 1.5$. As seen from Table 2, eight groups of curve parameters are found, and the estimated curve parameters are rather close to the true parameters.

Curve Number	Parameter
1 True	$(0.202\ s, 2.7027\ 10^{-4}\ s/m)$
1 Estimated	$(0.200\ s, 2.705\ 10^{-4}\ s/m)$
2 True	$(0.202\ s, 6.2500\ 10^{-4}\ s/m)$
2 Estimated	$(0.200\ s, 6.261\ 10^{-4}\ s/m)$
2 Estimated	$(0.204\ s, 6.245\ 10^{-4}\ s/m)$
3 True	$(0.232\ s, 2.5000\ 10^{-4}\ s/m)$
3 Estimated	$(0.232\ s, 2.503\ 10^{-4}\ s/m)$
4 True	$(0.232\ s, 6.2500\ 10^{-4}\ s/m)$
4 Estimated	$(0.232\ s, 6.245\ 10^{-4}\ s/m)$
4 Estimated	$(0.236\ s, 6.245\ 10^{-4}\ s/m)$
5 True	$(0.902\ s, 5.0000\ 10^{-4}\ s/m)$
5 Estimated	$(0.904\ s, 4.982\ 10^{-4}\ s/m)$
6 True	$(0.932\ s, 4.5455\ 10^{-4}\ s/m)$
6 Estimated	$(0.932\ s, 4.530\ 10^{-4}\ s/m)$
6 Estimated	$(0.932\ s, 4.546\ 10^{-4}\ s/m)$
7 True	$(1.302\ s, 2.7027\ 10^{-4}\ s/m)$
7 Estimated	$(1.300\ s, 2.721\ 10^{-4}\ s/m)$
7 Estimated	$(1.300\ s, 2.737\ 10^{-4}\ s/m)$
8 True	$(1.302\ s, 2.5000\ 10^{-4}\ s/m)$
8 Estimated	$(1.304\ s, 2.456\ 10^{-4}\ s/m)$
8 Estimated	$(1.304\ s, 2.472\ 10^{-4}\ s/m)$

Table 2: The results of the FCE-algorithm along with the true curve parameters. The width of the cluster band is set to 3, i.e. $\alpha = 1.5$.

B. *Example 2*

To demonstrate the performance of the FCE-algorithm in the presence of noise, an example based on a noise corrupted synthetic CMP-gather is given. The subsurface model is a pure acoustic (compressional waves only) horizontally layered subsurface consisting of four finite layers and infinite top and bottom layers. The synthetic CMP-gather is produced by use of ray tracing [10]. The resulting CMP-gather is composed of 29 reflection curves, shown in Figure 13, where each vertical line represents a trace and the signal values are illustrated using grey scale. The parameters for both image and parameter domains are given in Table 3.

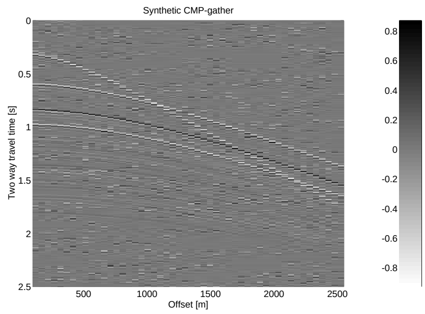

Figure 13: Synthetic CMP-gather composed of 29 reflection curves, added with uncorrelated noise.

Image domain		Parameter domain	
Par.	Value	Par.	Value
M	50	S	200
N	625	Z	625
$\Delta\kappa$	$50\ m$	$\Delta\sigma$	$1.57 \cdot 10^{-6}\ s/m$
Δt	$4 \cdot 10^{-3} s$	Δt_0	$4 \cdot 10^{-3} s$
κ_{min}	$100\ m$	σ_{min}	$4.0 \cdot 10^{-4}\ s/m$
t_{min}	$0\ s$	$t_{0,min}$	$0\ s$
α	1.5	λ_1	0.5
		λ_2	0.5
		v_σ	4
		v_{t_0}	2

Table 3: Image and parameter domain settings.

In Figure 14 the corresponding true reflection curve parameters are shown, that is the values of t_0 and σ corresponding to the 29 reflection curves.

Next, the multipulse technique [12] is used to map the seismic signals in the CMP-gather into a binary image, shown in Figure 15. As seen, the mapping process results in erroneous reflections, i.e., reflections without correlation over traces. Furthermore, the reflection curves have gaps, intersect and in cases are very close.

These problems are caused by the fact that the multipulse technique is a one dimensional process.

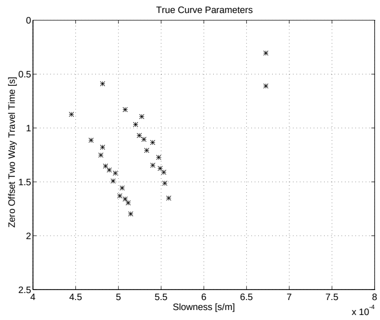

Figure 14: True reflection curve parameters.

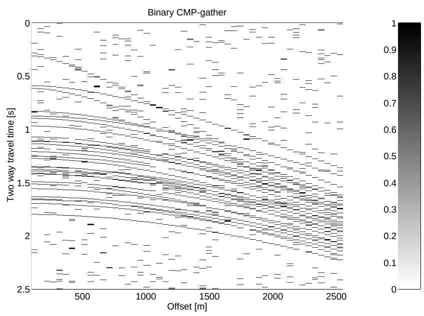

Figure 15: Binary CMP-gather obtained using multipulse technique at Figure 13.

Applying the FCE-algorithm to the binary image gives the IPM estimated parameter domain shown in Figure 16, using $v_{t_0} = 2$ and $v_\sigma = 4$.

Applying GRT on the parameter domain region specified by the resampled and thresholded IPM estimated parameter domain, with $\lambda_1 = 0.5$, results in the GRT parameter domain shown in Figure 17.

Finally, Figure 18 shows the estimated curve parameters, where a significance level of $\lambda_2 = 0.5$ in the final threshold process and $\alpha = 1.5$ in the clustering process have been used. Comparing Figure 18 with Figure 14, which shows the true reflection curves, all reflection curves are seen to have been found, and the accuracy of the estimated reflection curves is good. The deviations of the curve parameters are less than $\pm 2 \cdot 10^{-3}$ s for t_0 and $\pm 1.5 \cdot 10^{-6}$ s/m for σ.

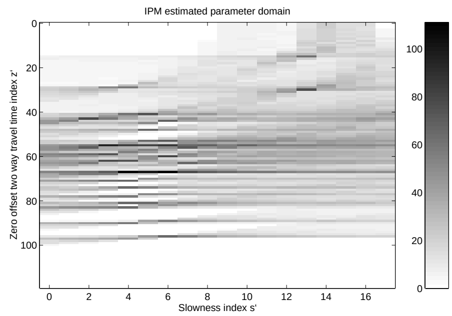

Figure 16: IPM estimated parameter domain.

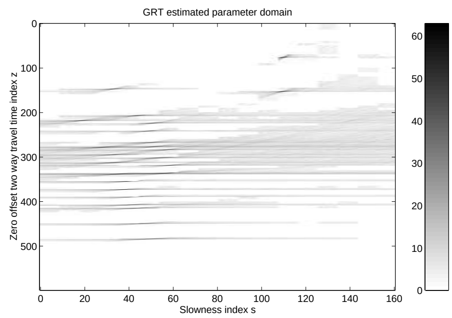

Figure 17: GRT estimated parameter domain using a pre-condition map, estimated by use of IPM.

VIII Conclusion

A new algorithm for fast curve parameter estimation, named the FCE-algorithm, has been presented. The algorithm identifies curve parameters by operating on a binary image, obtained, e.g., by edge filtering or deconvolution.

The fundamental idea of the algorithm is the use of pre-conditioning to reduce the computational cost of the traditional generalized Radon transform. The pre-conditioning map determ-

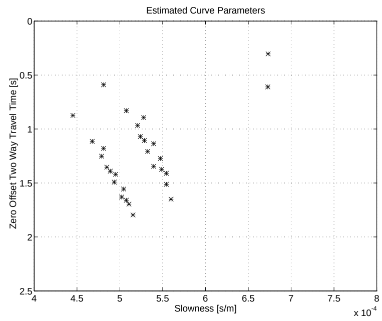

Figure 18: Estimated curve parameters using the FCE-algorithm.

ines regions of the parameter domain which contain peaks, and the generalized Radon transform is applied only in these regions. As the size of the regions is less than the full parameter domain, the pre-conditioning map reduces the computational costs when applying the generalized Radon transform. For fast generation of the pre-conditioning map, a generalization of the Hough transform named *image point mapping* has been developed. Image point mapping is computationally efficient by taking account of image points with value zero. The required parameter domain sampling and the resulting parameter domain blurring have been investigated.

The FCE algorithm was successfully applied to the identification of hyperbolas in seismic images and two numerical examples have been presented. One example demonstrates the potential of the algorithm for fast and accurate parameter estimation, and the other example illustrates the robustness of the algorithm with noise.

IX Acknowledgments

We wish to thank Peter Koefoed Møller, Katja Blankensteiner, Peter Skjellerup, and Jan Larsen for useful comments on this paper.

This work was partly supported by the Danish Academy of Technical Sciences and ØD-S Holding A/S.

A Clustering Using Hyperbolic Transformation Curves

In the case of hyperbolic transformation curves the exact shape of the parameter domain blurring can be calculated.

Let q denote a binary variable, $q \in \{+1, -1\}$. Thus, the lowest and highest curve within the image band specified by Equation 37 can be written

$$\sqrt{t_{0,0}^2 + (\sigma_0 \ \kappa)^2} + q \ \alpha \ \Delta t = \sqrt{t_0^2 + (\sigma \ \kappa)^2} \ \Leftrightarrow$$
$$t_0^2 = \left(\sqrt{t_{0,0}^2 + (\sigma_0 \ \kappa)^2} + q \ \alpha \ \Delta t\right)^2 - (\sigma \ \kappa)^2$$

It can be shown that κ_{low} and κ_{high} give the cluster limiting elliptical curves. The parameter cluster corresponding to the center curve $(t_{0,0}, \sigma_0)$ can be written as

$$\sqrt{\left(\sqrt{t_{0,0}^2 + (\sigma_0 \ \kappa_{low})^2} - \alpha \ \Delta t\right)^2 - (\sigma \ \kappa_{low})^2} < t_0$$
$$< \sqrt{\left(\sqrt{t_{0,0}^2 + (\sigma_0 \ \kappa_{low})^2} + \alpha \ \Delta t\right)^2 - (\sigma \ \kappa_{low})^2}$$

and

$$\sqrt{\left(\sqrt{t_{0,0}^2 + (\sigma_0 \ \kappa_{high})^2} - \alpha \ \Delta t\right)^2 - (\sigma \ \kappa_{high})^2} < t_0$$
$$< \sqrt{\left(\sqrt{t_{0,0}^2 + (\sigma_0 \ \kappa_{high})^2} + \alpha \ \Delta t\right)^2 - (\sigma \ \kappa_{high})^2}$$

Figure 19 (left) shows a hyperbolic center curve $(t_{0,0}, \sigma_0)$ with an uncertainty of $2\alpha\Delta t$. Figure 19 (right) shows the shape of the corresponding cluster, which is bounded by four elliptical curves. It should be noted that the cluster shape is highly dependent on the center curve $(t_{0,0}, \sigma_0)$.

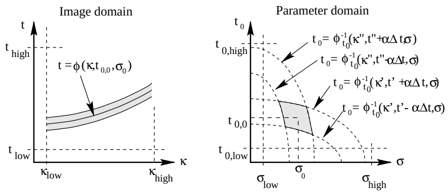

Figure 19: Left: Center curve with an uncertainty of $2\alpha\Delta t$ indicated. Right: Parameter cluster corresponding to curve band, where $\kappa' = \kappa_{low}$, $\kappa'' = \kappa_{high}$, $(t')^2 = t_{0,0}^2 + (\sigma_0\ \kappa_{low})^2$, and $(t'')^2 = t_{0,0}^2 + (\sigma_0\ \kappa_{high})^2$.

References

[1] P. V. C. Hough. A Method and Means for Recognizing Complex Patterns. US Patent: 3,069,654, Dec. 1962.

[2] R. O. Duda and P. E. Hart. Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Communication of the Association for Computing Machinery*, 15(1):11–15, 1972.

[3] J. Illingworth and J. Kittler. A Survey of the Hough Transform. *Computer Vision, Graphics, and Image Processing*, 44:87–116, 1988.

[4] H. Li, M. A. Lavin and R. J. Le Master. Fast Hough Transform: A Hierachical Approach. *Computer Vision, Graphics, and Image Processing*, 36:139–161, 1986.

[5] S. R. Deans. Hough Transform From the Radon Transform. *IEEE PAMI*, 3(2):185–188, 1981.

[6] J. Radon. Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten. *Ber. Ver. Sächs. Akad. Wiss. Leipzig, Math-Phys. Kl.*, 69:262–277, April 1917.

[7] S. R. Deans. *The Radon Transform and Some of Its Applications.* Krieger Publishing Company, Malabar, Florida, 2 edition, 1993. Previously John Wiley & Sons Inc., 1983.

[8] G. Beylkin. Discrete Radon Transform. *IEEE tr. ASSP*, 35(2):162–172, 1987.

[9] C. J. Haneveld and G. C. Herman. A Fast Algorithm for Computation of Radon Transforms. *Geophysical Prospecting*, 38:853–860, 1990.

[10] W. A. Schneider. The Common Depth Point Stack. *Proceesings of the IEEE*, 72(10):1238–1254, 1984.

[11] Özdoğan Yilmaz. *Seismic Data Processing.* Society of Exploration Geophysicists, Tulsa, Oklahoma, 1987.

[12] B. S. Atal and J. R. Remde. A New Model of LPC Excitation for Producing Natural-Sounding Speech at Low Bit Rates. In *Proceedings of ICASSP*, volume 1 of *ICASSP*, pages 614–617. IEEE, 1982.

[13] M. Cookey, H. J. Trussel and I. J. Won. Seismic Deconvolution by Multipulse Methods. *IEEE tr. ASSP*, 38(1):156–160, 1990.

[14] E. A. Robinson, T. S. Durrani and L. G. Peardon. *Geophysical Signal Processing.* Englewood Cliffs, New Jersey, Prentice-Hall Inc., 1980.

[15] J. Canny. A Computational Approach to Edge Detection. *IEEE tr. PAMI*, 8(6):679–698, 1986.

[16] K. Y. Huang, K. S. Fu, T. H. Sheen and S. W. Cheng. Image Processing of Seismograms: (A) Hough Transformation for the Detection of Seismic Patterns; (B) Thinning Processing in the Seismogram. *Pattern Recognition*, 18(6):429–440, 1985.

[17] C. Y. Chi, J. M. Mendel and D. Hampson. A Computationally Fast Approach to Maximum-Likelihood Deconvolution. *Geophysics*, 49(5):550–565, 1984.

Appendix I

Using the Generalized Radon Transform for Detection of Curves in Noisy Images

This appendix contains the paper *Using the Generalized Radon Transform for Detection of Curves in Noisy Images*, by Peter Toft.

The work has been presented at the *IEEE ICASSP 96* and this paper appears in the proceedings [3].

USING THE GENERALIZED RADON TRANSFORM FOR DETECTION OF CURVES IN NOISY IMAGES

Peter A. Toft

Electronics Institute, Technical University of Denmark, DK-2800 Lyngby.
Email ptoft@ei.dtu.dk. Phone +45 45934207

ABSTRACT

In this paper the discrete generalized Radon transform will be investigated as a tool for detection of curves in noisy digital images. The discrete generalized Radon transform maps an image into a parameter domain, where curves following a specific parameterized curve form will correspond to a peak in the parameter domain. A major advantage of the generalized Radon transform is that the curves are allowed to intersect. This enables a thresholding algorithm in the parameter domain for simultaneous detection of curve parameters. A threshold level based on the noise level in the image is derived. A numerical example is presented to illustrate the theory.

1. INTRODUCTION

In recent years the Hough transform [1] and the related Radon transform [2] have received much attention. These two transforms are able to transform two dimensional images with lines into a domain of possible line parameters, where each line in the image will give a peak positioned at the corresponding line parameters. This have lead to many line detection applications within image processing, computer vision, and seismics.

A natural expansion of the Radon transform is the (discrete) generalized Radon transform (GRT) [3, 4, 5]. Analogous to the linear Radon transform, the GRT transforms curves in the image into a discrete multi dimensional parameter domain producing peaks positioned at the corresponding curve parameters. In this way the GRT converts a difficult global detection problem into a more easily solved local peak detection problem. A major advantage of the GRT is that curves are allowed to intersect. Another major advantage that will be demonstrated in this paper, is that the GRT is very robust to noise.

In this paper a probabilistic approach is used to show that the GRT can be used for curve detection if the noise in the image is below a certain level compared to the signal values on the curves. If noise is added to an image containing curves, the problem is that peaks in the parameter domain may or may not correspond to actual curve parameters. A threshold level, based on the noise level, is derived and applied for separation of noise and curve information in the parameter domain. A numerical example is provided to illustrate the presented theory.

2. THE GENERALIZED RADON TRANSFORM

The Generalized Radon transform, GRT, of a digital image can be defined in many ways. One way is

$$\breve{g}(\boldsymbol{j}) = \sum_{l=0}^{L-1} g(\phi_m(l,\boldsymbol{j}), \phi_n(l,\boldsymbol{j})) \tag{1}$$

where $\breve{g}$ denote the GRT of the image $g(m,n)$ and $\boldsymbol{j}$ is a multi dimensional vector containing the curve parameters. The two curve functions $\phi_m(l,\boldsymbol{j})$ and $\phi_n(l,\boldsymbol{j})$ define the curve type and are (in principle) arbitrary[1]. A popular choice is the linear curve functions; e.g., normal parameters $\boldsymbol{j} = (\rho, \theta)$. Another frequent choice is the (τ, p)-parameters (known as slant stacking in seismics), where $\phi_m(l,\tau,p) = l$ and $\phi_n(l,\tau,p) = p\,l + \tau$.

Even though the GRT can be applied to any given image, the main feature of the GRT is that an image, which contains a discrete curve matching the curve functions at one parameter vector $\boldsymbol{j}^*$ implies that the parameter domain $\breve{g}(\boldsymbol{j})$, will show a peak at that specific parameter vector $\boldsymbol{j} = \boldsymbol{j}^*$. The linearity of the GRT implies that each curve in the image will be transformed into a peak in the parameter domain. In this paper a curve in an image is defined by large image values of the same sign on the curve and otherwise (approximately) zero.

Initially only two values of the GRT will be considered. The first, $\breve{g}(\boldsymbol{j}^*)$, corresponds to a curve in the

[1] An interpolation scheme is assumed implicitly; e.g., by rounding the functions $\phi_m(l,\boldsymbol{j})$ and $\phi_n(l,\boldsymbol{j})$ to the nearest sample point.

image. Another, $\breve{g}(\boldsymbol{j}^-)$, corresponds to a parameter vector, that does not match a curve in the image. It is assumed that $\breve{g}(\boldsymbol{j}^*)$ is the sum of a mean signal value μ over all L samples, and $\breve{g}(\boldsymbol{j}^-)$ covers (approximately) no samples of the curve(s) in the image. Both values of $\breve{g}$ are contaminated with noise due to noise in the image. Assume the noise in the image is nearly uncorrelated with zero mean (e.g., by subtracting a DC-value from the image) and variance σ^2.

A classical curve detection algorithm is to determine the parameter vectors from the positions of peaks in the parameter space

$$\boldsymbol{j}^* = \arg \{|\breve{g}(\boldsymbol{j})| \geq L\ \mu^*\} \tag{2}$$

The reason for choosing the significance level in this way is that Eq. 1 consists of a summation over L samples, and μ^* is a lower positive bound on the mean signal level on the curve; e.g, found by estimation. The purpose of the following is to estimate whether curves having the signal level μ can be detected using Eq. 2, if the image is contaminated with the described noise.

Due to the linearity, $\breve{g}$ consists of a curve part and a noise part. If $L \gg 1$ the sum of the noise terms $\breve{g}_{\text{noise}}$ will approximately be Gaussian distributed with zero mean and variance $L\sigma^2$ due to the Central Limit Theorem. This implies that the two considered values of the GRT are distributed as $\breve{g}(\boldsymbol{j}^*) = \mu L + \breve{g}^*_{\text{noise}} \in \mathcal{N}(\mu L, L\sigma^2)$ and $\breve{g}(\boldsymbol{j}^-) = \breve{g}^-_{\text{noise}} \in \mathcal{N}(0, L\sigma^2)$. Since Eq. 2 selects the large values in the parameter domain, an important issue is the probability of detecting the correct parameter vector of the two considered

$$\mathrm{P}_{\text{det 2}} \equiv \mathrm{P}\{|\breve{g}(\boldsymbol{j}^*)| > |\breve{g}(\boldsymbol{j}^-)|\} \tag{3}$$

Assuming that $\breve{g}(\boldsymbol{j}^*)$ and $\breve{g}(\boldsymbol{j}^-)$ are independent, inserting the joint probability distribution function, i.e, the product of the two individual Gaussian probability distribution functions, and using the trick of rotating the coordinate system 45 degrees, the integrals separate into one dimensional integrals which easily can be expressed by the error-function, erf(·)

$$\mathrm{P}_{\text{det 2}} = \frac{1}{2}\left[1 + \left(\mathrm{erf}\left(\frac{\lambda}{2}\right)\right)^2\right] \quad \text{and} \quad \lambda = \frac{\mu\sqrt{L}}{\sigma} \tag{4}$$

Note that in this case $\mathrm{P}_{\text{det 2}}$, shown on Fig. 1, only depends on one parameter λ. Note that $|\lambda| > 4$ gives an almost certain detection. This is the case if $\sigma \to 0$ or $\mu \to \infty$.

When using the GRT to detect curves then the discrete parameter domain will not only have two, but $\mathcal{J}$ different parameter vectors, where $\mathcal{J}$ is the number of samples in the parameter domain. It is assumed that all the noise sources in the parameter domain are independent and in the following, the detection of a single curve is analyzed. Selecting the position of the highest peak in the parameter domain, the probability of the selected parameter vector being correct, can be approximated by

$$\mathrm{P}_{\text{det all}} \cong \prod_{i=2}^{\mathcal{J}} \mathrm{P}_{\text{det 2}} \cong \mathrm{P}_{\text{det 2}}^{\mathcal{J}-1} =$$
$$\left(\frac{1}{2}\left[1 + \left(\mathrm{erf}\left(\frac{\lambda}{2}\right)\right)^2\right]\right)^{\mathcal{J}-1} \cong 1 - \frac{2\mathcal{J}}{\lambda\sqrt{\pi}}\,\mathrm{e}^{-\lambda^2/4} \tag{5}$$

The last simple approximation is valid if the detection probability is close to 1 as seen from Fig. 2. Several characteristics can be noted: The Figure shows a narrow transition from low to high detection probability as a function of λ, and $\mathcal{J}$ does not change the shape of $\mathrm{P}_{\text{det all}}$ significantly. If demanding a high $\mathrm{P}_{\text{det all}}$ then Eq. 5 and Fig. 2 demonstrate that $\mathcal{J}$ should be held low; i.e., by reducing the number of samples in the parameter domain to a minimum. It should be noted that this will involve a compromise on the range of parameter vectors.

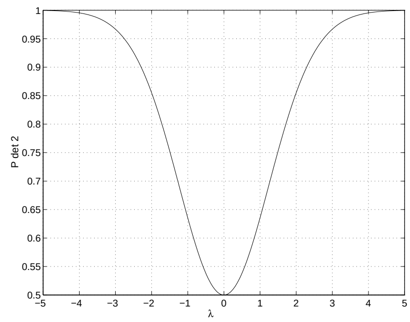

Figure 1: The probability of detecting the right curve parameters of two possible as a function of λ.

Eq. 5 can be used to set requirements on, e.g., the absolute mean signal level μ^* of the curve(s) to be detected. Demanding a detection probability $\mathrm{P}_{\text{det all}}$ greater than P^* implies that $\mu^* = \sigma/(\lambda^*\sqrt{L})$, where λ^* can be found by from Eq. 5 with a given detection probability, summation length L, number of samples in the parameter domain $\mathcal{J}$, and the standard deviation σ (e.g., by found by estimation). Any $|\mu|$ less than the threshold level, μ^*, can be considered as noise. In this way it is possible to give a statistically based estimate on the thresholding level in Equation 2.

Even though the above theory is developed by analyzing one curve in the image, the theory can be used

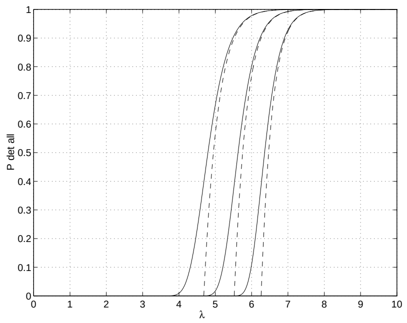

Figure 2: Solid lines show the probability of detecting the right curve parameter as a function of λ, and dashed lines show the simple approximation. From left to right $\mathcal{J}$=1000, $\mathcal{J}$=10000 and rightmost $\mathcal{J}$=100000.

if the image contains few curves. Instead of having one peak in the parameter domain representing one curve in the image, each of the Z curves in the image, where $Z \ll \mathcal{J}$, will give a peak in the parameter domain, even if the curves cross each other. With Z curves each of corresponding Z $\breve{g}$-values must be larger than the rest in the parameter domain. If only a few curves are present, the rest of the parameter domain is dominated by noise, and the probability of detection for each of the Z curves can be found from Eq. 5.

The theory used to derive Eq. 5 is somewhat pessimistic in estimation of the influence of the noise. This is partly due to the assumption that all the GRT-values include summing noise over L samples. Normally some of the GRT-values will require summing up over a curve partially outside the image, where the image must be assumed equal to zero. Furthermore some correlation must be expected in the parameter domain, especially if the number of dimensions in the parameter domain is higher than two. Depending on the sampling parameters, this implies that an effective $\mathcal{J}$ (less than the number of samples in the parameter domain) must be used in Eq. 5.

3. AN EXAMPLE OF LINE DETECTION IN A VERY NOISY IMAGE

A noise free image containing eight lines with limited slope is created. The image, shown in Fig. 3, has 101×101 samples. The curve sampling functions are chosen to $\phi_m = l - 50$ and $\phi_n = p\,(l-50) + \tau$ and L is set to 101. The offset is made in order to lower the sampling requirements in the parameter domain. The sampling distances in the parameter domain is set to $\Delta\tau = 1$ and $\Delta p = 0.01$. The line parameters are listed in Tab. 1.

No	p	τ	μ	No	p	τ	μ
1	0.10	60	1.5	5	0.00	30	1.0
2	0.25	50	-1.5	6	0.40	45	-1.0
3	-0.10	80	-1.0	7	-0.36	52	-1.0
4	-0.25	30	1.0	8	0.10	80	0.5

Table 1: Line parameters. p is the slope, τ is the offset, and μ is the curve amplitude.

To illustrate the potential of the GRT a very noisy image is generated, by adding Gaussian noise to the noise free image with zero mean and standard deviation $\sigma = 1$. It can be seen from Figure 4, that the lines are hard to identify. Choosing $\mathrm{P}_{\text{det all}} = 0.95$, Eq. 5 gives $\lambda^* = 6.53$; i.e., only lines with $|\mu| > 0.65$ should be detectable. This implies that all but line number eight should be detectable. The absolute value of the parameter domain obtained by the use of the GRT to the noisy image is shown in Fig. 5.

Since the noisy image contains few lines with absolute curve amplitude $|\mu|$ being of the same order of magnitude as σ and has approximately zero mean, σ was estimated from the image using the ordinary central variance estimator, which gave $\hat{\sigma} = 1.04$. Setting $\mathrm{P}_{\text{det all}}$ to 0.95, Eq. 5 results in $L\ \mu^* = 65.7$. This is used for thresholding of the parameter domain as shown in Figure 6. Seven of the eight line parameters are found despite the poor signal to noise ratio in the image. Note that some of the lines will be represented by a few neighbor parameter vectors. This error be can corrected by clustering neighbor parameter vectors. The error is due to the sampling of the parameter domain and the finite image size.

The theory predicted that only seven lines could be detected. The eighth line can be detected if the curve length can be increased or the noise variance can be reduced. If the theory is used with $\mathrm{P}_{\text{det all}}$ very low, $L\mu^*$ get lower and noise peaks will appear in the parameter domain along with parameters of the eighth line. In Fig. 7 the threshold level has been reduced to, e.g., $0.7L\mu^* = 46.0$. As it can be seen, noise will now give parameter vectors which do not represent a curve. As seen from Fig. 8 a further reduction of the threshold level to, e.g., $0.5L\mu^* = 32.9$ gives a parameter domain, where all eight lines are present. Due to the noise level many false parameter vectors can also be observed.

4. CONCLUSION

A statistically based noise analysis of the generalized Radon transform has been presented, which was used to derive a threshold level in order to separate curve information and noise in the parameter domain. A numerical example was provided to illustrate the theory.

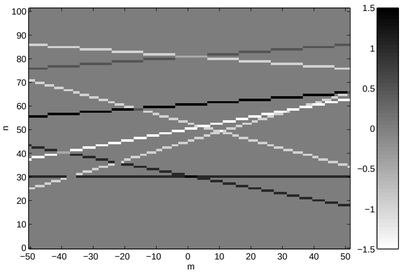

Figure 3: Noise free image with eight lines.

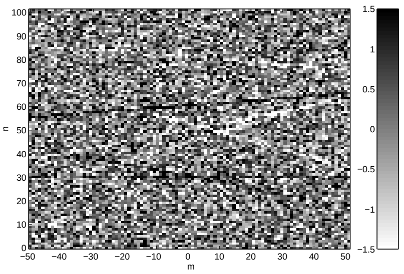

Figure 4: The same image contaminated with additive Gaussian noise.

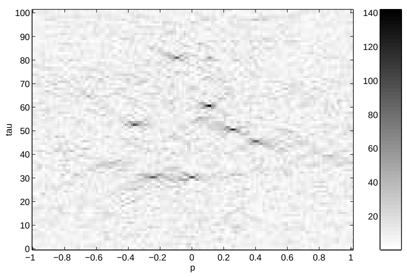

Figure 5: The absolute GRT of the noisy image. Note the peaks corresponding to the curves.

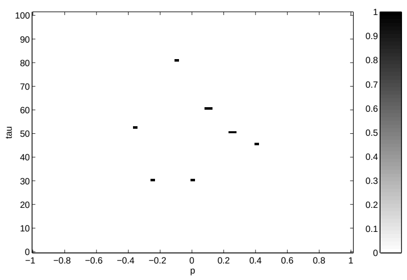

Figure 6: Threshold of the absolute GRT using the estimated threshold level.

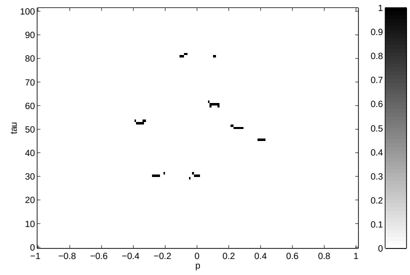

Figure 7: Threshold of the absolute GRT using 0.7 times the estimated threshold level.

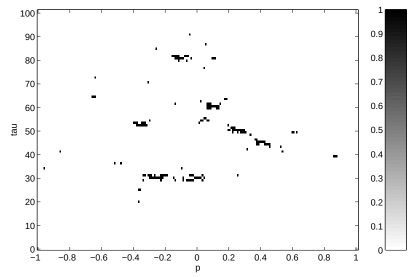

Figure 8: Threshold of the absolute GRT using 0.5 times the estimated threshold level.

5. REFERENCES

[1] J. Illingworth. A Survey of the Hough Transform. *Computer Vision, Graphics, and Image Processing*, 44, 1988.

[2] S. R. Deans. *The Radon Transform and Some of Its Applications.* Krieger Publishing Company, Malabar, Florida, 1993. Previously John Wiley & Sons Inc., 1983.

[3] G. Beylkin. Discrete Radon Transform. *IEEE ASSP*, 35(2):162–172, 1987.

[4] K. V. Hansen and P. A. Toft. Fast Curve Estimation Using Pre-Conditioned Generalized Radon Transform. Submitted for publication.

[5] P. A. Toft and K. V. Hansen. Fast Radon Transform for Detection of Seismic Reflections. In *Signal Processing VII - Theories and Applications*, volume I, pages 229–232. EURASIP EUSIPCO94, 1994.

Appendix J

Estimation of the Noise Contributions from Blank, Transmission and Emission Scans in PET

This appendix contains the paper *Estimation of the Noise Contributions from Blank, Transmission and Emission Scans in PET*, by Søren Holm, Peter Toft, and Mikael Jensen. The work has been carried out at the National University Hospital in Copenhagen in 1995.

The results have been presented at the *IEEE Medical Imaging Conference 95* and this paper will appear in the conference issue [7]. The abstract can also be found in the abstract collection of the MIC 95.

A more extensive article has also been submitted, see Appendix K.

Estimation of the Noise Contributions from Blank, Transmission and Emission Scans in PET

Søren Holm†, Peter Toft‡, and Mikael Jensen†
† Dept. of Nuclear Medicine, Rigshospitalet, National University Hospital, Copenhagen, Denmark (sholm@pet.rh.dk).
‡Electronics Institute, Technical University of Denmark (ptoft@ei.dtu.dk).

Abstract

This work determines the relative importance of noise from blank (B), transmission (T) and emission (E) scans in PET using a GE Advance scanner on a 20 cm cylinder, a brain phantom, and a torso-like ellipse (18/35 cm) with examples of human scans (brain O-15 water and F-18 FDG, heart FDG). Phantom E scans were acquired in both 2D and 3D modes as decay series with C-11 or F-18 over 3-6 decades of Noise Equivalent Counts (NEC). B and T scans were made using two pin sources (389+134 MBq) with times 16-32768 sec. In humans only a limited subset was available. In homogeneous phantoms normalized variance (var) was estimated from pixel distributions in single images. In other objects, including the human studies, calulations were performed on differences of paired images. In all cases a fit was made to a simple noise model. The cylinder data shows expected relations of T to B noise proving the adequacy of B scan times < 30 min for most purposes. For cylinder and brain phantom, contour plots are provided for var(E,T). In a typical 3D O-15 water study with 0.5M counts per central slice, a 10 min T-scan ensures that var(T)<0.1*var(E). Using 10 min T scan for a static 3D FDG brain study of 10 min having 5M counts yields equal E and T variance contributions. In human body scans T noise has a relative larger importance and is often dominated by effects of line artefacts from (clusters of) zeros in the T-scan, not included in the simple model.

I. Introduction

The purpose of this work is to establish guidelines for the relative importance of the noise contributions from blank, transmission and emission scans in typical imaging situations in PET. Although much information can be deduced theoretically from existing knowledge of reconstruction algorithms [1, 2], we have mainly applied an experimental approach. The limitations in generality caused by the obvious difficulty of varying all of the many possible parameters of acquisition and reconstruction are outweighed by the advantage of the close reproduction of actual scan setups, and direct testing in some human studies. The following paragraph describes the model used for fitting to the observed data.

II. Theory

Due to the the linearity of the reconstruction process the PET images (volumes) x_i, can be described as a weighted sum of the sinogram values s_j, with weight factors ϕ_{ij} corresponding to the actual (linear) reconstruction algorithm

$$x_i = \sum_j \phi_{ij} s_j \qquad (1)$$

where the actual image pixel (voxel) is denoted by i, and j denotes the pixel position in the sinogram. In order to compensate for attenuation the sinogram used for reconstruction is calculated from an emission sinogram e_j, a transmission sinogram t_j, and a blank scan b_j. The calculation basically amounts to:

$$s_j = \frac{e_j b_j}{t_j} \qquad (2)$$

although this is usually qualified by a filtering of the factor $\frac{b_j}{t_j}$.

All three types of sinograms in principle inherit their statistical properties from the Poisson statistics of the radioactive decay. A number of corrections that are performed before the stage of reconstruction described here, however, adds to the raw counts' noise. The overall effect of correction for randoms and scatter can be described as a decrease in effective counts [3], the resulting figure being widely known as Noise Equivalent Counts (NEC). A practical formulation used in this work is:

$$NEC = T\frac{(1-SF)^2}{1+2f\frac{R}{T}} \qquad (3)$$

where T is true counts (including scatter), R is random counts from the full field-of-view, f is the fraction of the sinogram covered by the object under investigation, and SF is the scatter fraction. In all applications of NEC in this study, the value of SF has been set to zero. Within this paper NEC per slice is used as a parameter.

A. *The Noise Model*

In the simplest case where the image noise is estimated from a single image set, using a Taylor expansion and disregarding noise correlation between pixels implies that the pixel noise variance $V\{x_i\}$ can be estimated by

$$\begin{aligned} V\{x_i\} &\simeq \sum_j \phi_{ij}^2 V\left\{\frac{e_j b_j}{t_j}\right\} \\ &\simeq \sum_j \phi_{ij}^2 E^2\{s_j\}\left[\frac{V\{e_j\}}{E^2\{e_j\}}+\frac{V\{b_j\}}{E^2\{b_j\}}+\frac{V\{t_j\}}{E^2\{t_j\}}\right] \quad (4) \end{aligned}$$

Due to the Poisson statistics the variance in the emission sinogram relative to the squared mean value $\frac{V\{e_j\}}{E^2\{e_j\}}$ is inversely proportional to the mean $E\{e_j\}$, i.e., inversely proportional to number of counts NEC in the emission sinogram. Similar arguments can be used for the transmission and blank scan terms, hence the total pixel variance normalized with the mean of the image can be modelled by

$$\frac{V\{x_i\}}{E^2\{x_i\}} = \frac{a}{NEC} + b + \frac{c}{T_t} + \frac{d}{T_b} \quad (5)$$

where T_t is the transmission scan time in seconds, T_b the blank scan time in seconds (for the given geometry and source strength). In the model a base term b is included to accommodate effects in the reconstruction process, not explained by the other terms, e.g., varying sensitivity of the detectors, not properly corrected for by normalization.

In this paper all measured data have been fitted to the model shown in Equation 5. In cases with heterogeneous ROI's the variance was measured for differences between equivalent and independent images [6]. In these cases the measured normalized variance has been divided by two so estimated noise parameters are directly comparable.

III. Methods

Measurements were made on the GE Advance scanner [8, 9] with 3D acquisition and reconstruction capability. Its 9 GByte raw data disk can hold about 180 3D frames (byte mode, separate prompts and delayed), convenient for phantom decay studies. Reconstruction in our configuration takes approximately 8 minutes per frame. The scanner applies two pin sources for blank and transmission scanning. During the initial experiments the pin source activities were 389 MBq and 134 MBq, respectively, and the blank scan sinogram count rate was 0.91 Mcps. On later phantom and human studies, blank and transmission scan times have been scaled in accordance with the currently observed blank scan count rates and the values quoted in seconds (powers of 2) are therefore assumed to be directly comparable.

A. *Phantom studies*

Three different water-filled phantoms have been used: the 20 cm standard (NEMA) cylinder, a torso-like ellipse with axes 18 and 35 cm, and an axially symmetrical, elliptical brain phantom (Capintec) with axes 15 and 20 cm.

Using initially the two homogeneous phantoms in accordance with Equation 5, emission scans were made in both 2D and 3D as decay series with F-18 starting at rates well below saturation of the scanner. In 2D a total of 22 measurements were performed starting with a half-life of F-18 and then reducing the time down to ...,4,2,1 seconds, effectively reducing the number of counts by a factor of 2 for each acquisition. In 3D the dataset was limited to 10 time frames. Transmission scans were made starting from T_t=16 seconds (64 on the ellipse) and doubling the time for each step up to 32768 seconds. Blank scans were made from T_b=16 to 4096 seconds and complemented with one 100000 seconds scan.

Reconstructions were made for all relevant combinations of emission and transmission scans using the 2048 seconds blank scan. All emission scans were further reconstructed with a calculated attenuation correction (CAC) using a circular or elliptical contour respectively. For the 2D cylinder case the emission image with highest count was reconstructed with the longest transmission scan and all blank scans. The cylinder was reconstructed in a 128^2 matrix with pixel size 2.0 mm, and a Hann filter (designated '4 mm' corresponding to the approximate resulting resolution FWHM). For the ellipse, a 128^2 matrix of pixel size 4.0 mm was used with a Hann '8 mm' filter. The 3D reconstruction further contains an axial filtering (Hann) set to its minimum value of 8.5 mm. The attenuation data were preprocessed with a Gaussian filter, 8 mm for the cylinder case, and 10 mm for the ellipse. All available corrections were applied, including detector normalization, randoms subtraction, scatter correction, (slice) sensitivity and absolute calibration, and decay correction to scan start. In all slices of all reconstructed images (order of 25,000 images) the mean and variance was calculated from an ROI extending 70% of the diameter or ellipse axes. These ROI-data were imported to MATLAB for further analysis. In 2D the data were averaged over 31 (of 35) slices avoiding edge effects but ignoring the minor differences between direct and cross slices. In 3D only the central 15 slices having almost identical noise were included in the average. For each reconstructed dataset the noise was represented by the normalized variance, i.e., variance/mean^2, in subsequent plots and fitting. From the total rates curves acquired with the images a Noise Equivalent Count (NEC) value per slice was calculated for each scan frame. The rate dependent correction between trues and NECs due to randoms was in all these cases below 15% except in the first 3D cylinder frame where it amounted to 25%.

The brain phantom has two separate chambers ('grey' and 'white' matter, respectively). It was filled to resemble the usually quoted ratio of 4:1 between these two substances for flow or metabolism. During scanning the phantom was embedded in a Rando-Alderson whole body phantom (normally used for radiotherapy dosimetry purposes) from which a few slices of the head had been removed. Also the thorax was replaced by the body ellipse phantom containing 5 times the total activity in the brain to provide more natural scatter and randoms conditions than the brain phantom alone. Two different sets of measurements have been performed. One set (repeated in 2D and 3D) was originally designed to address count rate performance [4]. The phantoms were loaded with C-11 Carbonate well above the expected saturation limit of the scanner and pairs of (60:62) seconds scans were performed every 10 minutes for about 5 hours (15 half lives). The relatively short half life of C-11 and the intention of having not only similar counts but also count rates in each of the two paired scans limited the maximum number of slice counts observed to 10^6 which would not allow a sufficiently clear distinction between the transmission scan times. One more decay series therefore was prepared in 3D with F-18 starting below maximum counts, and measuring for 12 half lives utilizing all the time for recording. Each of the half lives were split in two frames with a ratio of (0.415:0.585) yielding approximately equal counts. The C-11 series were reconstructed using an 'infinite' (32768 second) transmission scan. For the F-18 series, a set of 7 pairs of transmission scans were obtained (32,64,128,256,1024,4096,16384) seconds. Reconstructions were made to provide datasets with independent transmission and emission noise by combining the E-T frames as odd-odd, even-even. Reconstruction parameters for the brain phantom were identical to those for the cylinder except that the axial Hann filter was here replaced by a ramp filter. From both the C-11 series and the F-18 series, difference images were calculated, whole brain ROI's were placed in the original as well as the difference images for mean and standard deviation calculation respectively.

B. *Human studies*

For one person (case KL) included in a count rate performance and dose optimization study with O-15 [4] the usual 90 second integration time (starting 20 seconds after bolus injection) was supplemented by 2 short frames (20-22 seconds). The study comprised injections with 25-800 MBq in 2D and 50-800 MBq in 3D. Reconstruction and analysis was performed as above, although only with one 12 minutes transmission scan. The number of NECs in the short scans were about 16% of the corresponding 90 seconds frames normally used.

One person (case MN) injected with 240 MBq of FDG had a double set of emission scans (8-64 sec in 2D and 8-512 sec in 3D) of the brain starting 2 hours after injection, followed by a double set of transmission scans (32-512 sec) yielding a total of 35 data points in (E,T). Reconstruction and data analysis matched the brain phantom F-18 series.

One patient (case BB) undergoing a dynamic FDG scanning of the heart additionally had a series of transmission scans, (0.5,1,2,4,8,20) minutes. Pairs of emission scans (0.5,1,2,5 minutes) from different parts of the dynamic study were reconstructed with all the transmission scans with subsequent data analysis in a large body circumferential ROI.

IV. Results

In this section data and parameters are presented for the previously described experiments and compared to the model.

A. *The Cylinder and Elliptical Body phantom*

Estimated parameters for the Cylinder phantom in 2D and 3D modes, and the Elliptical Body phantom in 2D and 3D are given in table 1 and 2.

Table 1: Cylinder data 2D and 3D

	Cylinder 2D	Cylinder 3D
a [M Counts]	4.77E-2	2.70E-2
b []	1.88E-3	3.49E-4
c [sec]	6.40	1.64
d [sec]	0.96	
Min(NEC) [M-Counts]	0.43	1.48
Max(NEC) [M-Counts]	13.00	38.75
Min(T_t) [sec]	64	256
Max(T_t) [sec]	32768	32768
Min($\frac{V}{E^2}$)	5.60E-3	1.10E-3
Max($\frac{V}{E^2}$)	1.60E-1	2.46E-2
Max(Error)	1.26E-4	3.02E-6

Table 2: Elliptical body phantom data 2D and 3D

	Ellipse 2D	Ellipse 3D
a [M Counts]	2.46E-2	3.38E-2
b []	1.67E-3	1.54E-3
c [sec]	8.85	4.16
Min(NEC) [M-Counts]	0.17	2.50
Max(NEC) [M-Counts]	9.60	27.83
Min(T_t) [sec]	1024	1024
Max(T_t) [sec]	16384	32768
Min($\frac{V}{E^2}$)	4.01E-3	2.79E-3
Max($\frac{V}{E^2}$)	1.56E-1	1.94E-2
Max(Error)	4.77E-5	9.40E-6

In the following two Figures of the normalized variance are shown as a function of NEC and transmission scan length T_t. Figure 1 corresponds to the Cylinder (2D) and Figure 2 to the Cylinder (3D).

Also in the Brain phantom (3D) case and in case MN both the emission and transmission parameters have been estimated. The parameters are listed in Table 3 together with the range of the fit. Figure 3 shows the normalized variance for the Brain phantom and Figure 4 demonstrates that the model fits the measured data nicely.

To illustrate how well the emission data measured on brains agree with the phantom studies, Figure 5 shows case KL (2D/3D), case MN (2D/3D), and the Brain Phantom (2D/3D). In all cases data corresponding to only one transmission scan are shown.

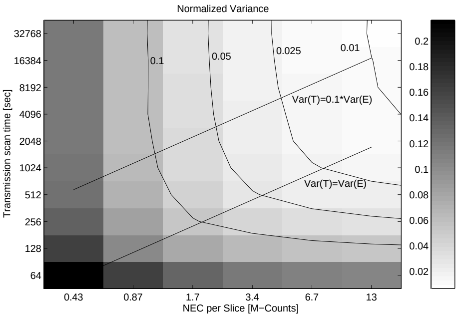

Figure 1: Normalized variance of Cylinder (2D) as a function of NEC per slice and duration of transmission scan T_t. Contours show iso-noise curves with levels indicated. Lines show where the ratio of transmission to emission noise terms is unity and 10% respectively.

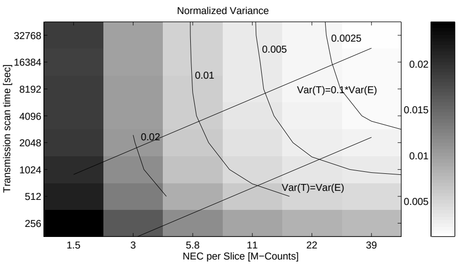

Figure 2: Normalized variance Cylinder (3D) as a function of NEC per slice and duration of transmission scan T_t..

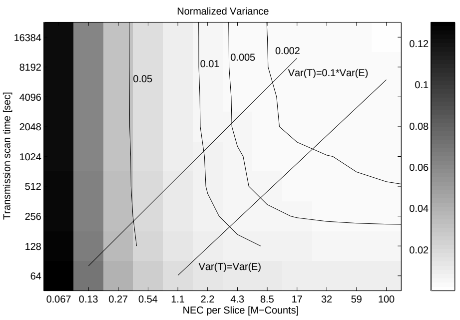

Figure 3: Normalized variance of Brain phantom (3D) as a function of NEC per slice and duration of transmission scan T_t.

Table 3: Brain phantom data 3D and Case MN 3D

	Brain Ph.	Case MN
a [M Counts]	1.67E-2	1.85E-2
c [sec]	1.01	1.59
Min(NEC) [M-Counts]	6.7E-2	6.4E-1
Max(NEC) [M-Counts]	2.2	3.6
Min(T_t) [sec]	64	64
Max(T_t) [sec]	16384	512

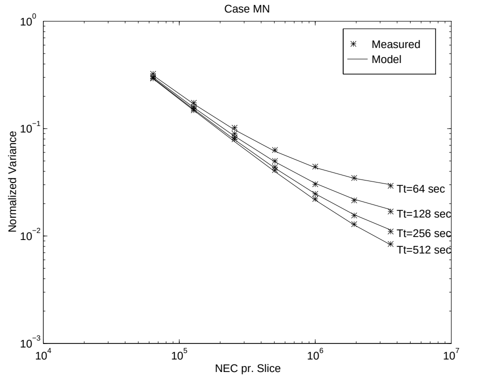

Figure 4: Normalized variance of case MN (3D) as a function of NEC per slice and duration of transmission scan T_t. Stars show the measured data and lines the fitted model.

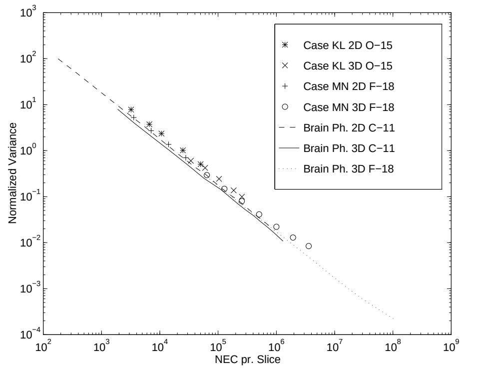

Figure 5: Normalized variance of Case KL (2D/3D), case MN (2D/3D), and Brain phantom (2D/3D) as a function of NEC per slice.

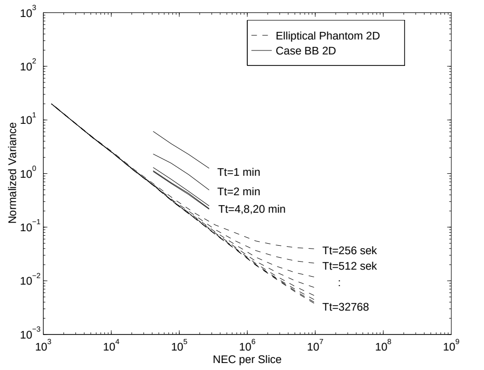

Figure 6: Normalized variance of body studies case BB (2D) and Elliptical phantom (2D) as a function of NEC per slice. Note that case BB (1,2 and 4 min) shows a different noise behavior.

Figure 6 shows body data. The 2D Elliptical Body phantom follows the previous noise model, but in case BB (large patient with arms in FOV) the noise curves for the 1 and 2 minutes transmission scans are shifted upwards. Inspection of the reconstructed images reveals large number of lines, due to (clusters of) zeros in the transmission scan.

V. Discussion

The general noise properties of reconstructive tomography are well described in the literature [1, 2]. Previous descriptions of the Advance scanner also include count rate dependence in phantoms and humans [8, 4] but the noise characterization so far [5] focused on the emission noise. In this paper we have determined the relative importance of the noise contributions from blank, transmission, and emission scans for a range of imaging situations encountered.

As shown, a good fit to the empirical variance is found both in 2D and 3D studies. The numerical values of the measure chosen for the noise examination is strongly dependent on reconstruction parameters, in part because of the neighbor pixel correlation disregarded in the theory section. Most parameters, however, will affect the Emission and Transmission contributions in the same way (through ϕ_{ij} only), and therefore the model and its output is considered adequate for the purpose of identifying areas in the E-T plane where one source is dominating.

From Table 1 it is seen that the blank scan contribution to the noise as expected is only a small fraction of the transmission for same duration, the ratio being well explained by the average attenuation of the (central part of) the 20 cm Cylinder phantom. For all practical purposes, therefore, the application of a 20 minutes blank scan will ensure that the blank scan contribution is negligible since the cases where a longer transmission scan might be applied are those with a higher attenuation.

A typical 3D brain activation study with O-15 water in our setup will contain 0.5 Mcounts/central slice, Figure 5,[4]. With a 10 minutes transmission scan, the transmission variance calculated from the phantom data in Table 3 is 1.67E-3 compared to the emission contribution of 3.34E-2, i.e., the transmission adds (only) 5% to the final result. For the human case MN the corresponding figures would be 2.65E-3 (transmission), 3.7E-2 (emission) and 7%, respectively. A 10 minutes, 2D FDG brain scan typically would also have 0.5-1 M counts yielding the same 5-10% ratio, while acquiring the same data in 3D would make the two contributions almost equal and therefore call for a more detailed analysis. It should be noted that in this imaging condition no attribute has been made to the additional noise from the emission correction of the transmission scan which is necessary if the transmission is performed with activity present.

Figure 6 suggests that within the emission count range of a typical dynamic heart FDG scan, the transmission noise contribution (from a 20 minutes transmission scan) is not dominating. Data are, however, not available that can demonstrate the region in which the transmission noise curves split up. It should be emphasized that the observed shift of the short-time curves is due to line-artefacts caused by (clusters of) zeros in the transmission sinogram. This deteriorating effect is usually more important and calls for improved method of attenuation correction, e.g., by using image segmentation as described in [7].

References

[1] A. E. Todd-Pokropek and P. H. Jarrit. The noise characteristics of SPECT systems. In P. J. Ell and B. L. Holman, editor, *Computed Emission Tomography*, pages 361–389. Oxford University Press, Oxford, 1982.

[2] N. M. Alpert et al. Estimation of the Local Statistical Noise in Emission Computed Tomography. *IEEE Trans. Med. Imag.*, MI-1(2):142–146, 1982.

[3] S. C. Strother et al. Measuring PET Scanner Sensitivity: Relating Count Rates to Image Signal-to-Noise Ratios Using Noise Equivalent Counts. *IEEE Trans. Nucl. Sci.*, NS-37(2):783–788, April 1990.

[4] S. Holm et al. 3D PET activation studies with $H_2{}^{15}O$ bolus injection. Count rate performance and dose optimization. In *Quantification of Brain Function using PET*. Academic Press, San Diego, 1995. In press.

[5] S. Pajavic et al. Noise Properties of 3D PET Images. *J. Nucl. Med.*, 36:p. 105 (abstract), 1995.

[6] S. R. Cherry et al. Improved Detection of Focal Cerebral Blood Flow Changes Using Three-Dimensional Positron Emission Tomography. *J. Cereb Blood Flow Metab*, (4):630–638, 1993.

[7] S. R. Meikle et al. Attenuation Correction Using Count-Limited Transmission Data in Positron Emission Tomography. *J. Nucl. Med.*, 34:143–150, 1993.

[8] T. K. Lewellen et al. Investigations of the Count Rate Performance of General Electric Advance Positron Emission Tomograph. *IEEE Trans. Nucl. Sci.*, NS-42(4):1051–1057, August 1995.

[9] T. R. DeGrado et al. Performance Characteristics of a Whole-Body PET Scanner. *J. Nucl. Med.*, 35(8):1398–1406, 1994.

Appendix K

Estimation of the Noise Contributions from Blank, Transmission and Emission Scans in PET

This appendix contains the paper *Estimation of the Noise Contributions from Blank, Transmission and Emission Scans in PET*, by Søren Holm, Peter Toft, and Mikael Jensen. The work has been carried out at the National University Hospital in Copenhagen in 1995.

The results have been presented at the *IEEE Medical Imaging Conference 95* and this paper has been submitted for publication in a conference issue of the IEEE Transactions on Nuclear Science [8].

A subset of this paper appears in [7], which is shown in Appendix J.

Estimation of the Noise Contributions from Blank, Transmission and Emission Scans in PET

Søren Holm†, Peter Toft‡, and Mikael Jensen†
† Dept. of Nuclear Medicine, Rigshospitalet, National University Hospital, Copenhagen, Denmark (sholm@pet.rh.dk).
‡Department of Mathematical Modelling, Technical University of Denmark (ptoft@ei.dtu.dk).

Abstract

This work determines the relative importance of noise from blank (B), transmission (T) and emission (E) scans in PET using a GE Advance scanner on a 20 cm cylinder, a brain phantom, and a torso-like ellipse (18/35 cm) with examples of human scans (brain O-15 water and F-18 FDG, heart FDG). Phantom E scans were acquired in both 2D and 3D modes as decay series with C-11 or F-18 over 3-6 decades of Noise Equivalent Counts (NEC). B and T scans were made using two pin sources (≈500 MBq total) over 64-32768 sec. In humans only a limited subset was available. In homogeneous phantoms normalized variance (var) was estimated from pixel distributions in single images. In other objects, including the human studies, calulations were performed on differences of paired images. In all cases a fit was made to a simple noise model. The cylinder data show expected relations of T to B noise proving the adequacy of B scan times ≤ 20 min for most purposes. For the brain phantom, a contour plot is provided for var(E,T). In a typical 3D O-15 water study with 0.5M counts per central slice, a 10 min T-scan adds less than 10% to the total noise level. An example shows how to split a total scan time between E and T scans, in order to minimize the variance.

I. Introduction

PET scans require correction for attenuation in order to be quantitative. Even in brain activation studies where absolute units are often replaced by relative values (i.e., normalized to a mean value), inter-individual comparisons still assume that values across an image reflect the local tracer concentration. Corrections can be applied, e.g., by assuming uniform values of attenuation within geometrically defined boundaries or in segmented areas of the image, but a more accurate description of the attenuation requires measurement by a transmission scan. Unfortunately this also adds noise to the image and adds to the total procedure time. In some cases, mainly in body scanning, the increase in noise is immediately noticed and may even be so high that quantification must be sacrificed in order to provide a reasonable visual impression. However, in other situations, effects that are invisible for the human observer might still be of importance when examining the subtle differences of brain activation by statistical methods. The purpose of this work is to establish guidelines for the relative importance of the noise contributions from blank, transmission, and emission scans in typical imaging situations. Although much information can be deduced theoretically from existing knowledge of reconstruction algorithms [1, 2, 3] we have here applied an experimental approach. The limitations in generality caused by the obvious difficulty of varying all of the many possible parameters of acquisition and reconstruction are outweighed by the advantage of the close reproduction of actual scan setups, including some human studies. The following Section describes the model used for fitting to the observed data. It should not be considered as a rigorous derivation; it merely lists the different structures of the datasets included in the study comprising homogeneous as well as inhomogeneous objects.

II. Theory

Due to the the linearity of the reconstruction process the PET images (volumes) x_i, can be described as a weighted sum of the sinogram values s_j, with weight factors ϕ_{ij} corresponding to the actual (linear) reconstruction algorithm

$$x_i = \sum_j \phi_{ij} s_j \tag{1}$$

where the actual image pixel (voxel) is denoted by i, and j denotes the pixel position in the sinogram. In order to compensate for attenuation the sinogram used for reconstruction is calculated from an emission sinogram e_j, a transmission sinogram t_j, and a blank scan b_j. The calculation basically amounts to:

$$s_j = \frac{e_j b_j}{t_j} \tag{2}$$

although usually qualified by a filtering of the factor $\frac{b_j}{t_j}$.

All three types of sinograms in principle inherit their statistical properties from the Poisson statistics of the radioactive decay. A number of corrections that are performed before the stage of reconstruction described here, however, adds to the raw counts' noise. The overall effect of correction for randoms and scatter can be described as a decrease in effective counts [4, 3], the resulting figure being widely known as Noise Equivalent Counts (NEC). A practical formulation used in this work is:

$$\mathrm{NEC} = \mathcal{T}\frac{(1-\mathrm{SF})^2}{1+2f\frac{\mathcal{R}}{\mathcal{T}}} \tag{3}$$

where $\mathcal{T}$ is true counts (including scatter), $\mathcal{R}$ is random counts from the full field-of-view, f is the fraction of the sinogram covered by the object under investigation, and SF is the scatter fraction. In all applications of NEC in this study, the value of SF has been set to zero.

While NEC as a global measure may be adequate for the noise description in the case of homogeneous phantoms extending the full axial field-of-view (AFOV), objects of limited extension may be better characterized by a value per slice. In the following, all NEC values quoted correspond to a single slice.

A. *One Emission and One Transmission Scan*

In the simplest case where the image noise is estimated from a single image set, using a Taylor expansion and assuming no noise correlation between pixels implies that the pixel noise variance $V\{x_i\}$ can be estimated by

$$\begin{aligned} V\{x_i\} &\simeq \sum_j \phi_{ij}^2 V\left\{\frac{e_j b_j}{t_j}\right\} \\ &\simeq \sum_j \phi_{ij}^2 E^2\{s_j\}\left(\frac{V\{e_j\}}{E^2\{e_j\}}+\frac{V\{b_j\}}{E^2\{b_j\}}+\frac{V\{t_j\}}{E^2\{t_j\}}\right) \end{aligned} \tag{4}$$

Due to the Poisson statistics the variance in the emission sinogram relative to the squared mean value $\frac{V\{e_j\}}{E^2\{e_j\}}$ is inversely proportional to the mean of an infinite amount of experiments $E\{e_j\}$, i.e., inversely proportional to number of counts NEC in the emission sinogram. Similar arguments can be used for the transmission and blank scan terms, hence the total pixel variance normalized with the mean of the image can be modelled by

$$\frac{V\{x_i\}}{E^2\{x_i\}} = \frac{a}{\mathrm{NEC}} + b + \frac{c}{T_t} + \frac{d}{T_b} \tag{5}$$

where T_t is the transmission scan time in seconds, T_b the blank scan time in seconds (for the given geometry and source strength). In the model a base term b is included to accommodate effects in the reconstruction process, not explained by the other terms, e.g., varying sensitivity of the detectors, not properly corrected for by normalization. For fixed values of (either) T_t or T_b the corresponding terms may also be thought of as part of this constant term. The model applies to regions with limited structural variation, hence measuring variance in a heterogeneous object like the brain or the thorax is problematic.

B. *Two Emission Scans*

To overcome the problem with heterogeneous structures the difference between two PET images, corresponding to two equivalent but independent emission scans $e_j^{(1)}$ and $e_j^{(2)}$, can be calculated. Due to Equations 1 and 2 the difference image is given by

$$x_i = \sum_j \phi_{ij}\left(e_j^{(2)} - e_j^{(1)}\right)\frac{b_j}{t_j} \tag{6}$$

Using a Taylor expansion, and using that the two emission scan have the same mean values $E\{e_j\}$ and variance $V\{e_j\}$ implies that

$$x_i \simeq \sum_j \phi_{ij}\left(\left(e_j^{(2)} - E\{e_j\}\right) - \left(e_j^{(1)} - E\{e_j\}\right)\right)\frac{E\{b_j\}}{E\{t_j\}} \tag{7}$$

Thus the variance $V\{x_i\}$ is given by

$$V_{\text{two e-scans}}\{x_i\} \simeq 2\sum_j \phi_{ij}^2 E^2\{s_j\}\frac{V\{e_j\}}{E^2\{e_j\}} \tag{8}$$

Normalized with the squared mean of the heterogeneous structure and using the same definition of constants as in Equation 5 gives that

$$\frac{1}{2}\frac{V_{\text{two e-scans}}\{x_i\}}{E^2\{x_i\}} \simeq \frac{a}{\mathrm{NEC}} \tag{9}$$

Thus only the emission term of the noise variance can be estimated from two emission scan and one transmission scan. Since all structural information is removed by using differences, no constant term (b) can appear.

If instead two transmission scans (and one emission scan) are available, it is easy to show that two times the transmission variance can be estimated (and only that) from a normalized difference image.

C. *Two Emission and Two Transmission Scans*

Using the same Taylor technique with two sets of emission scans and two transmission scans, it can be shown that

$$\frac{1}{2}\frac{V_{\text{two e and t-scans}}\{x_i\}}{E^2\{x_i\}} \simeq \frac{a}{\mathrm{NEC}} + \frac{c}{T_t} \tag{10}$$

This implies that both the emission and transmission term can be estimated in the inhomogenous case if a double set of independent and equivalent emission and transmission scans are measured and reconstructed pairwise. Note that any normalized variance shown in the following has been divided by two if measured from two scans, cf. Equations 5, 9 and 10.

D. *Zeroes in the Transmission Sinogram*

A common problem especially in body scanning is zeros in the transmission sinogram. One option is to replace the zeros with a small number T_0, so the attenuation correction can be done cf. Equation 2. Although overflow is avoided this may imply that reconstructed images contain strong lines causing a significant change in the noise structure. Let $\mathcal{Z}$ be the set of j, where the measured t_j is zero. Assume that the variance of the blank scan is negligible and further that only the terms with zeros effectively contribute to the noise. In this case

$$\begin{aligned} V\{x_i\} &\simeq \sum_{j\in\mathcal{Z}} \phi_{ij}^2 V\left\{\frac{e_j b_j}{T_0}\right\} \\ &\simeq \sum_{j\in\mathcal{Z}} \phi_{ij}^2 V\{e_j\}\frac{E^2\{b_j\}}{T_0^2} \end{aligned} \tag{11}$$

This indicates that the number of zeros (through the sum) and the variance on the emission scan determines the noise level. Hence the variance normalized with the squared mean is approximately proportional to the number of zeros and inversely proportional to NEC. In the limit where this kind of noise is dominating, however, the reconstructed images consist of many lines and become useless anyway.

E. *Limitations*

Using either of subsections A, B or C to estimate the noise terms, we estimate the noise variance from a chosen ROI and normalize with the squared mean of the ROI. Due to the global features in the reconstruction process the noise tends to have only a very small correlation with the underlying structure and often it is approximately evenly distributed across the ROI's used. This implies that that the normalized noise variance of the ROI depends strongly on the selected ROI, through the squared mean.

III. Methods

Measurements were made on the GE Advance scanner [5, 6, 7] with 3D acquisition and reconstruction capability. Its 9 GByte raw data disk can hold about 180 3D frames (byte mode, separate prompts and delayed), suitable for phantom decay studies. Reconstruction in our configuration (with 10 i860 processors) takes approximately 8 minutes per frame [8]. For blank and transmission scanning, which is always performed in 2D, the scanner applies two pin sources. During the initial experiments the pin source activities were 389 MBq and 134 MBq, respectively. The blank scan sinogram count rate was 0.91 Mcps, corresponding to an average count rate per sinogram element of $\approx$ 0.23 cps in the center. On later phantom and human studies, blank and transmission scan times have been scaled in accordance with the currently observed blank scan count rates and the values quoted in seconds are therefore directly comparable.

A. *Phantom studies*

Three different water-filled phantoms have been used: the 20 cm standard (NEMA) cylinder, a torso-like ellipse with axes 18 and 35 cm, and an axially symmetrical, elliptical brain phantom (Capintec) with axes 15 and 20 cm.

Using initially the two homogeneous phantoms in accordance with Equation 5, emission scans were made as decay series with F-18 in 2D (total of 22 measurements) and 3D (limited to 10 time frames). Transmission scans were made starting from T_t=64 seconds and doubling the time for each step up to 32768 seconds. Blank scans were made from T_b=64 to 4096 seconds and complemented with one 100000 seconds scan.

Reconstructions were made for all relevant combinations of emission and transmission scans using the 2048 seconds blank scan. All emission scans were further reconstructed with a calculated attenuation correction (CAC) using a circular or elliptical contour respectively. For the 2D cylinder case the emission image with highest count was reconstructed with the longest transmission scan and all blank scans. The cylinder was reconstructed in a 128^2 matrix with pixel size 2.0 mm, and a 4 mm Hann filter. For the ellipse, a 128^2 matrix of pixel size 4.0 mm was used with a 8 mm Hann filter. The 3D axial filter (Hann) was also set to its minimum value of 8.5 mm. The attenuation data were preprocessed with a 8 mm Gaussian filter for the cylinder case, and 10 mm for the ellipse. All available corrections were applied, including detector normalization, randoms subtraction, and scatter correction. In all slices of all reconstructed images (order of 25,000 images) the mean and variance was calculated from an ROI extending 70% of the diameter or ellipse axes. In 2D, the data were averaged over 31 (of 35) slices avoiding edge effects but ignoring the minor differences between direct and cross slices. In 3D only the central 15 slices having almost identical noise were included in the average. For each reconstructed dataset the noise was represented by the normalized variance, i.e., variance/mean^2, in subsequent plots and fitting. From the total rates curves a Noise Equivalent Count (NEC) value per slice was calculated for each scan frame. The rate dependent correction between trues and NECs due to randoms did not exceed 15%.

The brain phantom has two separate chambers ('grey' and 'white' matter, respectively). It was filled to resemble the usually quoted ratio of 4:1 between these two substances for flow or metabolism. Two different sets of measurements have been performed. One set (repeated in 2D and 3D) was originally designed to address count rate performance [9, 10]. The phantoms were loaded with C-11 carbonate well above the expected saturation limit of the scanner and pairs of (60:62) seconds scans were performed every 10 minutes for about 5 hours (15 half lives). The maximum number of slice counts observed in this study was limited to 10^6 which would not allow a sufficiently clear distinction between the transmission scan times. One more decay series was therefore prepared in 3D with F-18 measuring for 12 half lives, each split in two frames with a ratio of (0.415:0.585) yielding approximately equal NEC. The C-11 series were reconstructed using an 'infinite' (32768 sec) transmission scan. For the F-18 series, a set of 6 pairs of transmission scans, 64 - 16384 sec, were obtained. Reconstructions were made to provide datasets with independent transmission and emission noise according to Equation 10 by combining the E-T frames as odd-odd, even-even. Reconstruction parameters for the brain phantom were identical to those for the cylinder except that the axial Hann filter was replaced by a Ramp filter. From both the C-11 series and the F-18 series difference images were calculated, and ROI's placed over grey matter (central and peripheral), white matter, and whole brain in the original as well as the difference images for mean and standard deviation calculation respectively.

B. *Human studies*

For one person (case KL) included in a count rate performance and dose optimization study with O-15 [9], the usual 90 second integration time (starting 20 seconds after bolus injection) was supplemented by 2 short frames (20-22 sec). The study comprised of injections with 25-800 MBq in 2D and 50-800 MBq in 3D. Reconstruction and analysis was performed as above, although only with one 12 minutes transmission scan. The number of NECs in the short scans were about 16% of the corresponding 90 seconds frames normally used.

One person (case MN) injected with 240 MBq of FDG had a double set of emission scans (8-512 sec) of the brain starting 2 hours after injection, followed by a double set of transmission scans (64-512 sec) yielding a total of 28 data points in (E,T)-plane. Reconstruction and data analysis was made to match the brain phantom F-18 series.

One patient (case BB) undergoing a dynamic FDG scanning of the heart additionally had a series of 6 transmission scans (0.5 - 20 min). Pairs of emission scans (0.5,1,2,5 min) from different parts of the dynamic study were reconstructed with all the transmission scans, and subsequent data analysis made as described above (Equation 9) in a large body circumferential ROI.

C. *Fitting data*

From the emission data (see subsections in section II.) the parameter a was estimated by linear regression in a log-log diagram of the normalized variance using the longest transmission scan available. Depending on the setup the transmission parameter c (and the base term b) was subsequently estimated by subtracting the estimated emission part of the normalized variance, a/NEC, and again applying linear regression. This approach is very simple and was considered adequate for the purpose. Non-linear data fitting was also tried and found to give similar results.

IV. Results

In this section data and parameters are presented for the previously described experiments and compared to the model. Again it should be noted that any variance shown has been divided by two if measured from paired difference scans so that the results presented are comparable and represent noise in the single images.

Estimated model parameters from the Cylinder phantom in 2D and 3D modes, the Elliptical Body phantom in 2D and 3D, The Brain Phantom, and Case MN, are given in Table 1. This table give parameters valid for any practically used value of NEC and T_t.

Table 1 Estimated model constants.

Case	Mode	a [k counts]	b [ ]	c [sec]
Cylinder	2D	47.7	0.00188	6.40
Cylinder	3D	27.0	0.00035	1.64
Ellipse	2D	24.6	0.00167	8.85
Ellipse	3D	33.8	0.00154	4.16
Brain Ph.	3D	16.7		1.01
Case MN	3D	18.5		1.59

In Figure 1 the measured normalized variance in the Cylinder (2D) case is shown. It can be seen that the transmission scans will effectively start to add to the variance if NEC is larger than approximately 10^5 counts per slice and that they - for transmission scan lengths normally encountered - are dominating at 10^7. Note also, that the emission noise model as judged from the linearity in the log-log plot remains valid down to the level where the noise exceeds the mean by an order of magnitude, i.e., beyond any practical application of the images as such.

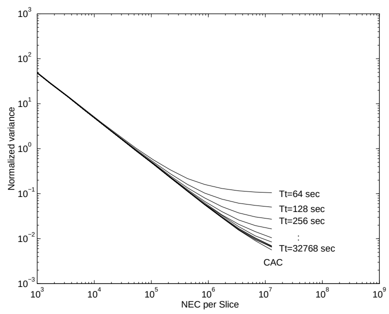

Figure 1 Normalized variance from 2D PET scan of the 20 cm Cylinder phantom as a function of NEC per slice and duration of transmission scan T_t. CAC means calculated attenuation correction.

In the Cylinder (2D) case the blank scan parameter d has also been estimated from reconstructed images corresponding to varying blank scan length T_b, and the estimated value is $d = 0.96$. By comparison with the parameter c from Table 1 the transmission scan in the cylinder case (2D) is seen to contribute approximately 6.7 times more to the variance than a blank scan with the same duration, in accordance with the average attenuation of the central region of a 20 cm waterfilled phantom.

In Figure 2, corresponding to the Brain phantom scanned in 3D, a contour plot shows the normalized variance as a function of NEC and transmission length T_t. Shading is used to indicate the actual variance, which can be seen from the rightmost grey scale bar. Contours show approximate equidistant level of noise. Furthermore two lines are inserted to show where the transmission term equals 10% and 100% respectively of the emission term.

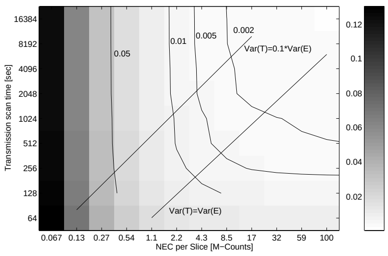

Figure 2 Normalized variance of Brain phantom (3D) as a function of NEC per slice and duration of transmission scan T_t. Contours show iso-noise curves with levels indicated. Lines show where the ratio of transmission to emission noise terms is unity and 10% respectively.

In the Brain phantom (3D) case and in case MN two emission and two transmission scans were measured, thus both emission and transmission parameters have been estimated. The parameters are listed in Table 1. Despite the approximations made for deriving the models Figures 3 and 4 demonstrate an excellent match between the measured data and the model for many decades of NEC. Note that the very low noise level, compared to Figure 1, is due to the use of difference images made from paired emission and transmission scans, which eliminates the b-term.

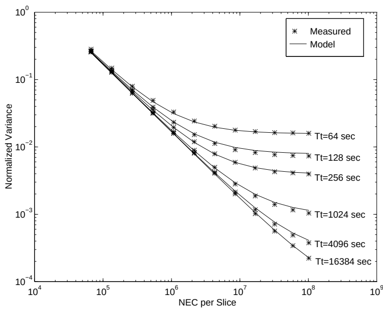

Figure 3 Normalized variance of Brain phantom (3D) as a function of NEC per slice and duration of transmission scan T_t. Stars show the measured data and lines the fitted model.

Figure 5 illustrates how well the emission data measured on human brains agree with the phantom studies. The Figure shows case KL (2D/3D), case MN (2D/3D), and the Brain Phantom (2D/3D). In all cases data corresponding to only one transmission scan are shown.

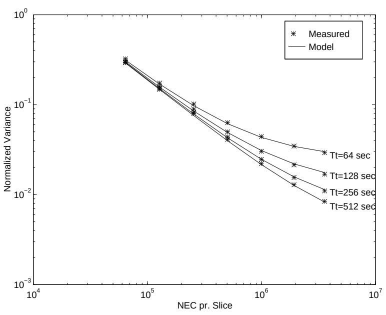

Figure 4 Normalized variance of case MN (3D) as a function of NEC per slice and duration of transmission scan T_t. Stars show the measured data and lines the fitted model.

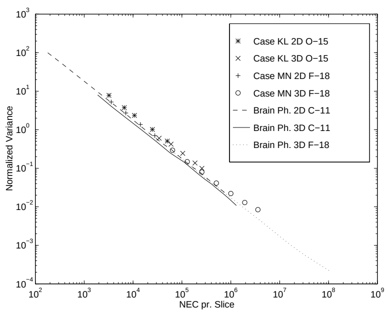

Figure 5 Normalized variance of Case KL (2D/3D), case MN (2D/3D), and Brain phantom (2D/3D) as a function of NEC per slice.

Table 2 shows noise estimates obtained from different ROIs. The emission parameter a and the transmission parameter c have been estimated for the Brain Phantom (3D) in four different ROIs. The Head ROI is an ellipse just surrounding the activity of the brain, WM is a small region within white matter, GM_p and GM_c are two small grey matter regions, peripheral and central, respectively. The averages of the measured regions are normalized to the average of WM. Note the approximately constant ratio between a and c, indicating the major influence of the average on the coefficients.

Table 2 Brain phantom data (3D) with different ROIs.

ROI	a [k counts]	c [sec]	Average
Head	17	1.0	2.07
WM	72	3.7	1.00
GM_p	4.4	0.24	3.98
GM_c	7.2	0.49	4.05

Finally Figure 6 shows data from body measurements. The 2D Elliptical Body phantom follows the previous noise model, but in case BB (large patient with arms in FOV, only a single set of transmission scans) the noise curves for the 1 and 2 minutes transmission scans are shifted upwards. Inspection of the reconstructed images reveals large number of lines, due to (clusters of) zeros in the transmission scan (cf. Equation 11).

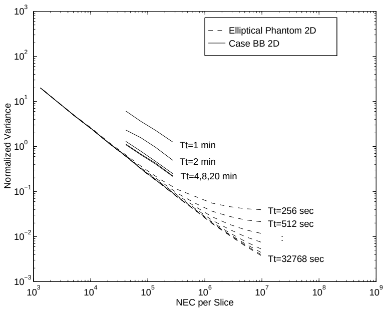

Figure 6 Normalized variance of body studies case BB (2D) and Elliptical phantom (2D) as a function of NEC per slice. Note that case BB (1,2 and 4 min) shows a different noise behavior.

V. Optimization

One application of the estimated noise parameters is optimization of the patient time used in the scanner, [3]. Assume that the total time available for the examination of a patient is T seconds and that the transmission scan and the emission scan are distinct. The duration of the two scans are T_t and T_e respectively, and $T = T_t + T_e$. For simplicity assume that the rate of Noise Equivalent counts is constant and equals R_{NEC}. This implies that the sum of the emisssion and transmission term are given by

$$\frac{V}{E^2} = \frac{a}{R_{\mathrm{NEC}}(T - T_t)} + \frac{c}{T_t} \tag{12}$$

Thus, with respect to the total noise level, the optimum duration of the transmission scan is given by

$$T_t = \frac{T}{1 + \sqrt{\frac{a}{c\, R_{\mathrm{NEC}}}}} \tag{13}$$

Figure 7 shows the normalized variance of the individual noise terms and their sum as a function of transmission scan time T_t. The Figure corresponds to the parameters found in Table 1 for the Brain Phantom (3D) and the sum of the transmission scan time and emission scan time is arbitrarily set to 1800 sec. In Figure 7 it is assumed that $R_{\mathrm{NEC}} = 1000$ counts per slice per sec, which gives an optimized normalized variance of 0.014 when $T_t = 354$ sec. Assuming $R_{\mathrm{NEC}} = 10000$ counts per slice per sec implies that the optimum is found at $T_t = 786$ sec and the normalized variance is 0.0029.

VI. Discussion

The general noise properties of reconstructive tomography are well described in the literature [1, 2]. Previous descriptions of the Advance scanner also include count rate dependence in phantoms and humans [6, 9, 10] but the

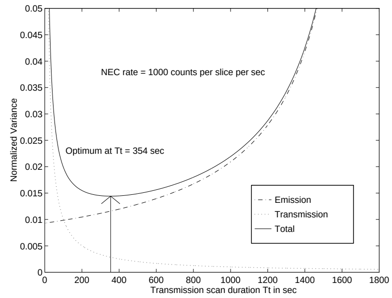

Figure 7 The transmission and emission noise terms and their sum as a function of the transmission scan duration, when the total (patient) scan time is 1800 sec. Parameters correspond to the Brain Phantom (3D). The rate of Noise Equivalent counts is 1000 counts per slice per sec.

noise characterization so far [11] focused on the emission noise. A theoretical derivation [3] using data from the GE 2048 scanner formed part of the basis for the design of the Advance scanner's 2-pin source transmission scanning system. In the present paper we have measured the relative importance of the noise contributions from blank, transmission, and emission scans for a range of imaging situations encountered. Transmission scan values are presented as scan durations in seconds for a given, specified set of sources. Optimal conditions with 2 sources of maximal strength (400 MBq) would almost double the transmission count rate. Given the current expensive Germanium-68 pin sources, the presented configuration represents a likely mean value over source life time. Results can be scaled according to the observed blank scan count rate, to accomodate for differences in number or activity of pin sources and their rotation radius.

An excellent fit to the empirical variance has been found both in 2D and 3D studies. The numerical values of the measure chosen for the noise examination is strongly dependent on reconstruction parameters, in part because of the neighbor pixel correlation disregarded in the theory section. Most parameters, however, will affect the emission and transmission contributions in the same way (through ϕ_{ij} only), and therefore the model and its output is considered adequate for the purpose of identifying areas in the E-T plane where one source is dominating. The analysis largely ignores the fact that the tomographic noise is non-stationary with higher values found towards the center as examplified by the differently located ROIs in the brain phantom, but also in this respect is the ratio between the emission and the transmission term approximately constant.

From the estimated value of $d = 0.96$ it is seen that the blank scan contribution to the noise as expected is only a small fraction of the transmission noise for same duration, the ratio being well explained by the average attenuation of the (central part of) the 20 cm Cylinder phantom. For all practical purposes, therefore, the application of a 20 minutes blank scan will ensure that the blank scan contribution is negligible since the cases where a longer transmission scan might be applied are those with a higher attenuation.

A typical 3D brain activation study with O-15 water in our setup will contain 0.5 Mcounts/central slice, (Figure 5,[9]). With a 10 minute transmission scan, the transmission variance calculated from the phantom data in Table 1 is 0.00167 compared to the emission contribution of 0.0334, i.e., the transmission adds (only) 5% to the final result. If difference images are made, the effect becomes even smaller. For the case MN the corresponding figures would be 0.00265 (transmission), 0.037 (emission) and 7%, respectively. This higher value is due to the lack of a skull for the Brain phantom.

A 10 minute, 2D FDG brain scan typically would also have 0.5-1 M counts yielding the same 5-10% ratio, while acquiring the same data in 3D would make the two contributions almost equal and therefore call for a more detailed analysis of timing. It should be noted that in this imaging condition no attribute has been made to the small additional noise from the emission correction of the transmission scan which is necessary if the transmission is performed with activity present.

Figure 6 suggests that within the emission count range of a typical dynamic heart FDG scan, the emission noise contribution (when using a 20 minute transmission scan) is dominating. Data are, however, currently not available that can demonstrate the region in which the transmission noise curves split up. It should be emphasized that the observed shifts of the curves with short transmission time are due to line-artifacts caused by (clusters of) zeros in the transmission sinogram. This deteriorating effect is usually more important and calls for an improved methods of attenuation correction, e.g., by using image segmentation as described in [12].

VII. Conclusion

The measured image noise variance from a GE Advance PET scanner has been modelled as a sum of terms corresponding to the noise in the emission scan, the transmission scan, and the blank scan. The weight parameters in this simple model have been determined from a large number of experiments in both 2D and 3D scan mode for the 20 cm standard (NEMA) cylinder, a torso-like ellipse with axes 18 and 35 cm, and an axially symmetrical, elliptical brain phantom (Capintec) with axes 15 and 20 cm. Furthermore, noise model parameters have been found from some human studies. For brain studies there is an excellent agreement between model and observation, and between phantoms and human studies. The estimated parameters have been used to optimize the durations of the emission and transmission scan under the constraint that their sum is constant. For studies of the heart or other body regions the model is still generally valid, but the parameters not as well documented, and the transmission noise is often dominated by line artifacts.

References

[1] A. E. Todd-Pokropek and P. H. Jarrit. The noise characteristics of SPECT systems. In P. J. Ell and B. L. Holman, editor, *Computed Emission Tomography*, pages 361–389. Oxford University Press, Oxford, 1982.

[2] N. M. Alpert et al. Estimation of the Local Statistical Noise in Emission Computed Tomography. *IEEE Trans. Med. Imag.*, MI-1(2):142–146, 1982.

[3] C. W. Stearns and D. Wack. A Noise Equivalent Counts Approach to Transmission Imaging and Source Design. *IEEE Trans. Med. Imag.*, MI-12(2):287–292, June 1993.

[4] S. C. Strother et al. Measuring PET Scanner Sensitivity: Relating Count Rates to Image Signal-to-Noise Ratios Using Noise Equivalent Counts. *IEEE Trans. Nucl. Sci.*, NS-37(2):783–788, April 1990.

[5] T. R. DeGrado et al. Performance Characteristics of a Whole-Body PET Scanner. *J. Nucl. Med.*, 35(8):1398–1406, 1994.

[6] T. K. Lewellen et al. Investigations of the Count Rate Performance of General Electric Advance Positron Emission Tomograph. *IEEE Trans. Nucl. Sci.*, NS-42(4):1051–1057, August 1995.

[7] T. Lewellen et al. Investigations of the Performance of the General Electric Advance Position Emission Tomograph in 3D mode. In *Submitted*, 1995.

[8] P. S. Crandall and C. W. Stearns. A Scalable Multi Processor Implementation of the Reprojection Algorithm for Volumetric PET Imaging. In *Conference Record of the IEEE 1995 Nuclear Symposium and Medical Imaging Conference.*, 1996. In press.

[9] S. Holm et al. 3D PET activation studies with $H_2{}^{15}O$ bolus injection. Count rate performance and dose optimization. In *Quantification of Brain Function using PET*. Academic Press, San Diego, 1996. In press.

[10] S. Holm et al. Count Rate Performance and Dose Optimization for rCBF with O-15 Water Bolus Injection in the New 3D PET. *J. Nucl. Med.*, 36:105P, 1995.

[11] S. Pajavic et al. Noise Properties of 3D PET Images. *J. Nucl. Med.*, 36:105P, 1995.

[12] S. R. Meikle et al. Attenuation Correction Using Count-Limited Transmission Data in Positron Emission Tomography. *J. Nucl. Med.*, 34:143–150, 1993.

Appendix L

A very fast Implementation of 2D Iterative Reconstruction Algorithms

This appendix contains the abstract and summary of the paper *A very fast Implementation of 2D Iterative Reconstruction Algorithms*, by Peter Toft and Jesper James Jensen [5]. The abstract and summary has been submitted to the *IEEE Medical Imaging Conference 96*. Note that a paper has also been submitted, see Appendix M.

A very fast Implementation of 2D Iterative Reconstruction Algorithms

Peter Toft† and Jesper James Jensen‡

† IMM, Technical University of Denmark, DK-2800 Lyngby, Denmark, Email ptoft@ei.dtu.dk.
‡ ØDS-Holding A/S, Kroghsgade 1, DK-2100 Copenhagen E, Denmark, Email jjj@oedan.dk.

abstract

One of the limitations of using iterative reconstruction methods in tomography is the slow performance compared with the direct reconstruction methods, such as filtered backprojection. In this paper we demonstrate a very efficient implementation of virtually all types of iterative reconstruction methods. The key idea of our methods is to generate the huge system matrix only once, and store it using sparse matrix techniques. From the sparse matrix we can perform the matrix vector products very fast, which implies a huge acceleration of the reconstruction. In this paper we demonstrate that iterative reconstruction algorithms can be implemented and run almost as fast as direct reconstruction algorithms. The method has been implemented in a software package which is available for free, providing ART, EM, and the Least Squares Conjugate Gradient Method.

A very fast Implementation of 2D Iterative Reconstruction Algorithms

Peter Toft† and Jesper James Jensen‡

† IMM, Technical University of Denmark, DK-2800 Lyngby, Denmark, Email ptoft@ei.dtu.dk.
‡ ØDS-Holding A/S, Kroghsgade 1, DK-2100 Copenhagen E, Denmark, Email jjj@oedan.dk.

1 Introduction

In the last 15 years the iterative reconstruction methods have gained much attention in the literature [1]. Several methods have been very prominent, such as EM (Expectation Maximization) [2, 3], ART (Algebraic Reconstruction Technique) [4], and LSCG (Least Squares Conjugate Gradient) [5]. Each of the methods formulates the reconstruction problem as a linear set of equations $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x} \Leftrightarrow b_i = \sum_{j=1}^{J-1} a_{i,j}x_j$, where $\boldsymbol{b}$ is an I-dimensional vector containing the known sinogram values wrapped into a vector, and $\boldsymbol{x}$ is a J-dimensional vector containing the unknown image to be reconstructed. The matrix $\boldsymbol{A}$ is the system matrix, which contains the weight factors between each of the image pixels and each of the sinogram values, corresponding to line orientations.

One problem is the huge size of the system matrix, often impossible to store in memory. Assuming that memory is not available for storing the full system matrix, then one possibility is to compute the individual matrix elements in each iteration when needed. This can be done by using of the discrete Radon transform or other modelling schemes of the scanner. This solution is rather easily implemented and is viable and storage requirements are reduced to a minimum, by only requiring memory for the sinogram ($\boldsymbol{b}$) and the current solution ($\boldsymbol{x}$), and perhaps some additional temporary variables of the same size or smaller, but no system matrix is stored in memory. This implementation has a major drawback in speed, due to the many times the system matrix has to be computed during an iterative reconstruction. Each time at the same high cost.

2 Accelerated 2D Iterative Reconstruction

Here a hybrid solution is proposed for accelerating the reconstruction speed of the iterative methods and only requiring as much memory as modern workstations are currently equipped with or will be soon. The idea is to store the non-zero elements of the system matrix in the memory using sparse matrix techniques. In this way the core of the reconstruction algorithms, highly based on matrix vector multiplications, can be accelerated significantly, and thereby solve one of the major drawbacks of the iterative methods.

It is proposed that the system matrix $\boldsymbol{A}$ is calculated one time only using all the modifications found for the actual scanner setup. If no specific scanner model is provided then the system matrix can be modelled and generated using the Radon transform or other simpler schemes. From the system matrix the very small values in the matrix are truncated to zero by using a very small threshold level.

The sparse structure of $\boldsymbol{A}$ can be exploited by only storing the non-zero values in the fast memory. For a certain row, number i, all of the matrix elements are calculated, stored, and truncated. Hereby the number of non-zero elements in the row, denoted by Z_i, will be much smaller than the image size $J = M^2$. The values of Z_i are stored in a simple one dimensional vector. Two vectors of length Z_i indexed by an integer z can then be allocated containing both the non-zero matrix value a_z and the corresponding column index j_z. Assuming that 4 bytes are required for storing each of the vector elements then the total storage requirement is reduced to approximately $8\sum_{i=1}^{I} z_i \approx 8IM$ bytes, (assuming a nearest neighbour approximation with one pixel for each point along the integration lines). Assuming that a sufficient amount of memory is present most iterative algorithms can be implemented from three basic operations: Matrix vector multiplication $\boldsymbol{A}\tilde{\boldsymbol{x}}$, scalar product between the i'th row of the system matrix and a vector $\boldsymbol{a}_i^T\tilde{\boldsymbol{x}}$, and finally multiplication with the transpose of the matrix $\boldsymbol{A}^T\tilde{\boldsymbol{b}}$.

A software package has been written in C including the proper structures for manipulating sparse matrices and vectors, along with an optimized code for computing matrix vector products, well suited for iterative reconstruction algorithms. In the package ART, EM, and LSCG have all been implemented in a fast version using sparse matrix storage of the system matrix and in a slow version where the system matrix is not stored and needed matrix entries are computed in each part of the iterative steps. The software package is provided for free, protected by the GNU General Public License.

Several interpolation methods have been implemented for generation of the system matrix: (1): Nearest Neighbour interpolation based on the Radon transform. (2): Linear interpolation based on the Radon transform. (3): Analytical Radon transform of a square, i.e., the length through a quadratic pixel is used. (4): The Radon transform of a sinc expansion in the image domain.

3 Results

The program has been used on two types of machines. A Linux machine with a 120 MHz Pentium processor and a SGI Onyx with four 200 MHz R4400 processors, with the program running on one processor.

As an example a sinogram with $125 * 101$ samples is reconstructed into an image with $101 * 101$ pixels. In Table 1 the reconstruction time is shown for the fast and the slow methods on the two machines.

Times are measured for ART, EM, and the LSCG-method, where EM and LSCG were running (arbitrarily) 20 iterations, and ART 20 full iterations, i.e., 20 times the number of rows (chosen randomly), which is 20∗125∗101 iterations. Note that all times only correspond to the actual iterations. For the fast versions of the iterative reconstruction algorithms, the time to generate the system matrix once should be added if changing the parameters determining the system matrix, e.g., the sampling parameters of the sinogram or the reconstructed image.

For this example the system matrix was modelled using discrete Radon transformation with linear interpolation, where the threshold was chosen to zero, hence the slow and the fast methods give exactly the same results. In this example the sparse system matrix required approximately 13 MBytes memory.

Note that the large difference in speedup between ART and EM/LSCG is due to a very effective implementation of the forward projection compared with the backprojection. The slow methods can be accelerated some by implementing multiplication with the transpose of the system matrix (adjoint operator) as a backprojection integral, but note that this implies that the approximation of the system matrix will be different in the forward and the backprojection part.

Machine	Fast ART	Slow ART	Fast EM	Slow EM	Fast LSCG	Slow LSCG
Pentium	26 sec	2218 sec	17 sec	5776 sec	17 sec	5250 sec
Onyx	16 sec	1306 sec	16 sec	2722 sec	16 sec	2715 sec

Table 1 Time usage for 20 iterations of EM and LSCG. For ART the time is for 20 full iterations, i.e., 20∗125∗101 iterations.

4 Conclusion

We have demonstrated a very fast implementation of iterative reconstruction based on storing the system matrix in fast memory by using sparse techniques. The approach is mainly applicable to 2D reconstruction, due to the requirements of a sufficient amount of memory, but in principle the method can also be applied to 3D reconstruction.

The cost of the proposed idea is that a large amount of memory is required, but for ART we have demonstrated a speedup factor of approximately 70 and for EM and LSCG a speedup factor larger than 175, between the slow and the fast implementation. In the example it was showed that each iteration of ART, EM, or LSCG requires approximately one second on most modern workstations.

References

[1] Yair Censor. Finite series-expansion reconstruction methods. In *Proc. of the IEEE*, volume 71, pages 409–419, March 1983.

[2] L. A. Shepp and J. B. Krustal. Computerized tomography: The new medical x-ray technology. *Am. Math. Monthly*, 85:420–439, April 1978.

[3] R. E. Carson and K. Lange. The EM parametric image reconstruction algorithm. *Journal of the American Statistical Association*, 389(80):20–25, March 1985.

[4] Gabor T. Herman and Lorraine B. Meyer. Algebraric reconstruction techniques can be made computationally efficient. *IEEE trans. on Med. Imag.*, 12(3):600–609, Sep. 1993.

[5] John A. Scales. Iterative methods for large, sparse, inverse calculations. Samizdat Press. Colorado School of Mines, 76 Olcott Drive, White River Junction, Vermont 05001, April 1993. Lecture notes.

Appendix M

Accelerated 2D Iterative Reconstruction

This appendix contains the paper *Accelerated 2D Iterative Reconstruction*, by Peter Toft and Jesper James Jensen, which has been submitted to IEEE Transactions on Medical Imaging [6]. A subset of this paper has been submitted to the Medical Imaging Conference 96. This paper is shown in Appendix L.

Accelerated 2D Iterative Reconstruction

Peter Toft, Department of Mathematical Modelling,
Technical University of Denmark, DK-2800 Lyngby, Denmark, Email ptoft@ei.dtu.dk.
Jesper James Jensen, ØDS-Holding A/S, Kroghsgade 1,
DK-2100 Copenhagen E, Denmark, Email jjj@oedan.dk.

Abstract

One of the limitations of using iterative reconstruction methods in tomography is the slow performance compared with the direct reconstruction methods, such as filtered backprojection. In this paper we demonstrate a very fast implementation of most types of iterative reconstruction methods. The key idea of our method is to generate the huge system matrix only once, and store it using sparse matrix techniques. From the sparse matrix we can perform the matrix vector products very fast, which implies a major acceleration of the reconstruction algorithms. In this paper we demonstrate that iterative reconstruction algorithms can be implemented and run almost as fast as direct reconstruction algorithms. The method has been implemented in a software package that is available for free, providing reconstruction algorithms using ART, EM, and the Least Squares Conjugate Gradient Method.

I. Introduction

In the last 15 years the iterative reconstruction methods have gained much attention in the literature [1, 2]. Several methods have been very prominent, such as EM (Expectation Maximization) [3, 4, 5, 6, 7], ART (Algebraic Reconstruction Technique) [8, 1, 9], and LSCG (Least Squares Conjugate Gradient) [10, 11].

These methods formulate the reconstruction problem as a linear set of equations

$$\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x} \;\Leftrightarrow\; b_i = \sum_{j=1}^{J-1} a_{i,j} x_j, \quad i = 1, 2, \ldots, I \qquad (1)$$

where $\boldsymbol{b}$ is an I-dimensional vector containing the known sinogram values wrapped into a vector, and $\boldsymbol{x}$ is a J-dimensional vector containing the unknown image to be reconstructed. Here $\boldsymbol{A}$ is the system matrix, which contains the weight factors between each of the image pixels and each of the values in the sinogram, corresponding to line orientations. Compared with Radon transform based direct reconstruction methods [12], the use of linear algebra has several advantages, such as easier incorporation of irregular geometries. The system matrix can model several real-world properties, such as finite, i.e., non-zero detector size and varying detector sensitivity. Furthermore regularization can easily be incorporated [13, 10] in order to affect the often ill-conditioned reconstruction problem.

One problem is the huge size of the system matrix. A 2D sinogram from, e.g., a GE Advance PET scanner contains $I = 281 * 336$ values, and reconstructed into a $J = 301 * 301$ grid, i.e., the system matrix has approximately 8.5 billion elements, requiring over 34 GBytes of memory, when using 4 bytes per matrix element. This is a large amount of memory, even looking some years into the future. Besides this aspect, it would not be wise to store all that data, due to the fact that approximately 98% of the matrix entries will be zeros. This knowledge should be incorporated into the reconstruction schemes.

Assuming that memory is not available for storing the full system matrix, one possibility is to compute the individual matrix elements in each iteration when needed. This can be done by using the Radon transform, e.g., [14] or other modelling schemes for the scanner. This approach is rather easily implemented and is viable and storage requirements are reduced to a minimum, only requiring memory for the sinogram ($\boldsymbol{b}$) and the current solution ($\boldsymbol{x}$), and perhaps some additional temporary variables of the same size or smaller, but no system matrix is stored in memory. It will be demonstrated that this implementation has a major drawback in speed, since the system matrix will be computed many times during an iterative reconstruction. Each time at the same high computational cost.

II. Accelerated 2D Iterative Reconstruction

Here a hybrid solution is proposed for accelerating the iterative reconstruction algorithms, but requiring as much memory as modern workstations are currently equipped with, or will be soon. The idea is to store the non-zero elements of the system matrix in the main memory using sparse matrix techniques. In this way the core of the reconstruction algorithms, highly based on matrix vector multiplications, can be accelerated significantly, and thereby solve one of the major drawbacks of the iterative methods.

It is proposed that the system matrix $\boldsymbol{A}$ is calculated one time only using all the modifications found for the actual scanner setup. If no specific scanner model is provided then the system matrix can be modelled and generated using the Radon transform or other simpler schemes. From the system matrix the very small values in the matrix are truncated to zero.

$$\tilde{a}_{i,j} = \begin{cases} a_{i,j} & \text{if } a_{i,j} > \gamma \\ 0 & \text{Otherwise} \end{cases} \tag{2}$$

where the threshold γ can be chosen to a certain fraction of the maximum matrix value, e.g., $\gamma = 0.05 \max_{i,j}\{a_{i,j}\}$. If γ is chosen sufficiently low, a good compromise between resolution and the sparseness of the matrix can be reached, and normally this does not alter the behaviour of the algorithms. Current work concerns the quantification of the truncation error.

The sparse structure of $\boldsymbol{A}$ can be exploited by only storing non-zero values in the fast memory. For a certain row, number i, all of the matrix elements are calculated, stored, and truncated using Eq. 2. Hereby the number of non-zero elements in the row, denoted by Z_i, will be much smaller than the image size $J = M^2$. The values of Z_i are stored in a simple, one dimensional vector. Two vectors of length Z_i, indexed by an integer z, can then be allocated and stored containing the non-zero matrix value a_z and the corresponding column index j_z. The procedure is repeated for all rows.

Assuming a nearest neighbour approximation with one pixel for each point along the integration lines and using 4 bytes for storing each of the vector elements, the total storage requirement is then reduced to approximately $8\sum_{i=1}^{I} z_i \approx 8IM$ bytes. In the example shown above approximately 100 MBytes memory is required. Assuming this amount of memory is present most iterative algorithms can be implemented from three basic operations: Matrix vector multiplication $\boldsymbol{A}\tilde{\boldsymbol{x}}$, scalar product between the i'th row of the system matrix and a vector $\boldsymbol{a}_i^T\tilde{\boldsymbol{x}}$, and finally multiplication with the transpose of the matrix $\boldsymbol{A}^T\tilde{\boldsymbol{b}}$.

In the following pseudo code (called Algorithms), the implementation of the matrix vector multiplication and the multiplication with the transpose of the system matrix are shown. A C++ style is used for comments and note that all indices here start at zero.

Algorithm 1 : $\boldsymbol{Ax}$

```
For i = 0 to I-1                  //For all rows
 sum = 0                          //Initialize
 Set a and j to correct row       //Use pointers
 For z=0 to Z(i)-1                //For row i
  sum=sum+a(z)*x(j(z))            //Increment sum
 End
 bt(i)=sum                        //Store value
End
```

End Algorithm

Algorithm 2 : $\boldsymbol{A}^T\boldsymbol{b}$

```
For j=0 to J-1                    //For all columns
 xb(j)=0                          //Initialize
End
For i=0 to I-1                    //For all rows
 Set a and j to correct row       //Using pointers
 For z=0 to Z(i)-1                //Compute sum
  xb(j(z))=xb(j(z))+a(z)*b(i)     //Update sum
 End
End
```

End Algorithm

III. Implemented Methods

A software package has been written in C including the proper structures for manipulating sparse matrices and vectors, along with an optimized code for computing matrix vector products, well suited for iterative reconstruction algorithms. In the package ART, EM, and LSCG are implemented both in a fast version using sparse matrix storage of the system matrix and in a slow version where the system matrix is not stored and needed matrix entries are computed in each step of the iterative algorithms. The software package is available for free, but protected by the GNU General Public License. Contact the authors to obtain the package.

ART For a certain row i of the matrix (i depends on the iteration number k), the general iteration step incrementing the current solution $\boldsymbol{x}^{(k)}$ can be found in, e.g., [1]

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \frac{b_i - \boldsymbol{a}_i^T \boldsymbol{x}^{(k)}}{\boldsymbol{a}_i^T \boldsymbol{a}_i} \boldsymbol{a}_i \tag{3}$$

EM The general iteration step of EM [5, 15] requires a forward projection, a backprojection, and two fast updates in each iteration.

$$\boldsymbol{b}^f = \boldsymbol{A}\boldsymbol{x}^{(k-1)} \tag{4}$$

$$b_i^r = \frac{b_i}{b_i^f} \tag{5}$$

$$\boldsymbol{x}^b = \boldsymbol{A}^T \boldsymbol{b}^r \tag{6}$$

$$x_j^{(k)} = \frac{x_j^{(k-1)} x_j^b}{s_j} \text{ where } s_j = \sum_{i=1}^{I} a_{i,j} \tag{7}$$

LSCG The Least Squares Conjugate Gradient method requires some initialization [11]

$$\boldsymbol{s}^{(0)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(0)} \tag{8}$$

$$\boldsymbol{r}^{(0)} = \boldsymbol{p}^{(0)} = \boldsymbol{A}^T \boldsymbol{s}^{(0)} \tag{9}$$

$$\boldsymbol{q}^{(0)} = \boldsymbol{A}\boldsymbol{p}^{(0)} \tag{10}$$

Then for each iteration the LSCG algorithm on the normal equations becomes

$$\alpha = \frac{\left(\boldsymbol{r}^{(k-1)}\right)^T \boldsymbol{r}^{(k-1)}}{\left(\boldsymbol{q}^{(k-1)}\right)^T \boldsymbol{q}^{(k-1)}} \tag{11}$$

$$\boldsymbol{x}^{(k)} = \boldsymbol{x}^{(k-1)} + \alpha \boldsymbol{p}^{(k-1)} \tag{12}$$

$$\boldsymbol{r}^{(k)} = \boldsymbol{A}^T \boldsymbol{s}^{(k-1)} \tag{13}$$

$$\boldsymbol{r}^{(k)} = \boldsymbol{r}^{(k-1)} - \alpha \boldsymbol{q}^{(k-1)} \tag{14}$$

$$\beta = \frac{\left(\boldsymbol{r}^{(k)}\right)^T \boldsymbol{r}^{(k)}}{\left(\boldsymbol{r}^{(k-1)}\right)^T \boldsymbol{r}^{(k-1)}} \tag{15}$$

$$\boldsymbol{p}^{(k)} = \boldsymbol{r}^{(k)} + \beta \boldsymbol{p}^{(k-1)} \tag{16}$$

$$\boldsymbol{q}^{(k)} = \boldsymbol{A}\boldsymbol{p}^{(k)} \tag{17}$$

For all three methods an initial value of the solution, i.e., $\boldsymbol{x}^{(0)}$ is needed. In the package an image found by, e.g., a fast direct method, can be supplied and used. If not provided, all of the initial values of the vector are initialized to a properly chosen constant.

Interpolation Methods: Several interpolation methods have been implemented for computing the system matrix.

- Nearest Neighbour interpolation based on the Radon transform.
- Linear interpolation based on the Radon transform.
- Analytical Radon transform of a square, i.e., the length through a quadratic pixel is used.
- The Radon transform of a sinc expansion in the image domain.

IV. Results

The program has been used on two types of machines. A Linux machine with a 120 MHz Pentium processor and an Onyx from SGI equipped with four 200 MHz R4400 processors, where the program was running on one processor.

A. *Example 1*

In the first example the (synthetic) sinogram has $125 * 101$ samples and the reconstructed image has $101 * 101$ samples. In Table 1 the reconstruction times on both machines are shown for the fast and the slow method as well as the ratio between the execution times (slow/fast).

Times are measured for ART, EM, and the LSCG-method, when EM and LSCG were running (arbitrarily) 20 iterations, and ART 20 full iterations, i.e., 20 times the number of rows (chosen randomly), which is 20∗125∗101 iterations in Eq. 3. Note that all times only correspond to the actual iterations. For the fast versions of the iterative reconstruction algorithms, the time to generate the system matrix once should be added if changing the system matrix, e.g., when changing the sampling parameters of the reconstructed image.

For this example the system matrix was modelled using discrete Radon transformation with linear interpolation, where the threshold γ was chosen to zero, hence the slow and the fast methods give exactly the same results. Note that the large difference in speedup between ART and EM/LSCG is due to the implementation of the forward projection is more efficient than the multiplication with the transpose of the system matrix. The slow methods can be accelerated some by implementing multiplication with the transpose of the system matrix (adjoint operator) as a backprojection integral, but note that this implies that the approximation of the system matrix will be different in the forward and the backprojection part. The sparse system matrix for this transformation geometry required approximately 13 MBytes, and each iteration requires approximately one second.

Machine	Type	ART	EM	LSCG
Pentium	Fast	26 sec	17 sec	17 sec
	Slow	2218 sec	5776 sec	5250 sec
	Ratio	85	340	309
SGI Onyx	Fast	16 sec	16 sec	16 sec
	Slow	1306 sec	2722 sec	2715 sec
	Ratio	82	170	170

Table 1 Time usage for 20 iterations of EM and LSCG. For ART the time is for 20 full iterations, i.e., 20∗125∗101 of the iterations used in Eq. 3. The time measurements are for a sinogram with $125 * 101$ samples reconstructed into a $101 * 101$ samples image.

B. *Example 2*

In Fig. 1 a 2D sinogram with 281∗336 samples is shown, which was measured on a GE Advance PET scanner. The sinogram is reconstructed into a (large) image with 301∗301 samples. The system matrix has 94416∗90601 elements of which 0.23% are non-zero when modelling then system matrix using the Radon transform of a square. On the Onyx it required 1466 seconds to generate the 157 MBytes sparse matrix, and each iteration of EM required 15 seconds in the fast version and 11678 sec in the slow implementation. After 10 iterations the algorithm was stopped. The reconstructed image is shown in Fig. 2. For sake of comparison, the same sinogram reconstructed image using Filtered Backprojection is shown in Fig. 3, and this reconstruction used 6 seconds on the Onyx.

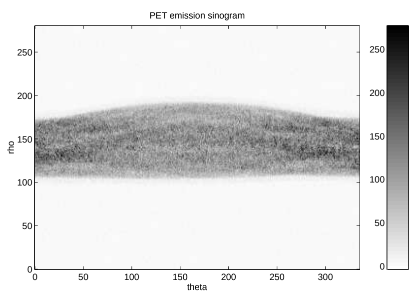

Figure 1 A PET sinogram of a human brain.

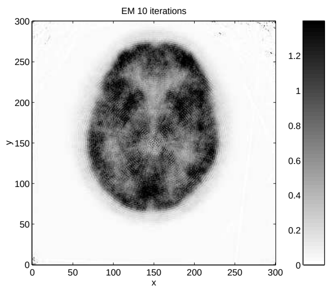

Figure 2 The reconstructed image after 10 iterations of EM.

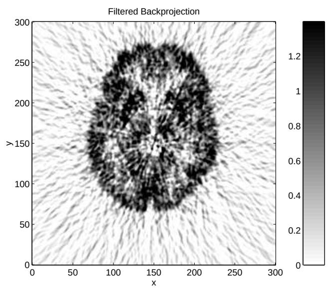

Figure 3 The reconstructed image using Filtered Backprojection with a ramp filter.

C. *Example 3*

Here the LSCG method is used to reconstruct a 101∗101 image from the sinogram shown in Fig. 1. On the Onyx 162 seconds is required to generate the system matrix and each iteration required 3 seconds. The reconstructed image after 11 iterations is shown in Fig. 4.

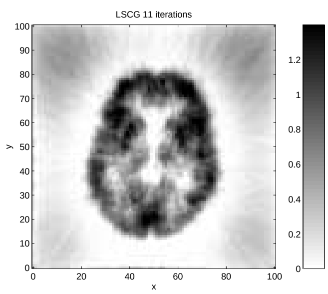

Figure 4 The reconstructed image using 11 iterations of the Least Squares Conjugate method. The artifacts around the brain can be removed by proper masking.

V. Conclusion

We have demonstrated a very fast implementation of iterative reconstruction comparable in speed with direct reconstruction methods. The implementation is based on storing of the system matrix in fast memory using sparse techniques. The approach is mainly applicable to 2D reconstruction, due to the requirements of a sufficient amount of memory, but in principle the method can also be applied to 3D reconstruction.

The cost is that a large amount of memory is required, but for ART we have demonstrated a speedup factor of approximately 80 and for EM and LSCG 170-340 depending on the machine, for a fixed transformation geometry and interpolation level. Once again these factors could be somewhat moderated by a faster implementation of the multiplication with the inverse system matrix.

Acknowledgment

We thank associate professor Lars Kai Hansen for support and useful comments.

References

[1] Yair Censor. Finite series-expansion reconstruction methods. In *Proc. of the IEEE*, volume 71, pages 409–419, March 1983.

[2] Julia K. Older and Paul C. Johns. Matrix formulation of computed tomography reconstruction. *Physics in Medicine & Biology*, 38:1051–1064, 1993.

[3] L. A. Shepp and J. B. Krustal. Computerized tomography: The new medical x-ray technology. *Am. Math. Monthly*, 85:420–439, April 1978.

[4] L. A. Shepp Y. Vardi and L. Kaufman. A statistical model for positron emission tomography. *Journal of the American Statistical Association*, 80(389):8–37, March 1985. The pages include comments and discussion by several authors.

[5] R. E. Carson and K. Lange. The em parametric image reconstruction algorithm. *Journal of the American Statistical Association*, 389(80):20–25, March 1985.

[6] Linda Kaufman. Implementing and accelerating the em algorithms for positron emission tomography. *IEEE Trans. Med. Imag.*, MI-6(1):37–51, march 1987.

[7] John M. Ollinger. Maximum-Likelihood reconstruction of transmission images in emission computed tomography via the EM algorithm. *IEEE Trans. Med. Imag.*, 13(1):89–101, March 1994.

[8] Gabor T. Herman and Lorraine B. Meyer. Algebraric reconstruction techniques can be made computationally efficient. *IEEE Trans. Med. Imag.*, 12(3):600–609, Sep. 1993.

[9] Huaiqun Guan and Richard Gordon. A projection access order for speedy convergence of art (algebraric reconstruction technique): a multilevel scheme for cumputed tomography. *Phys. Med. Biol.*, 39:2005–2022, 1994.

[10] Linda Kaufman and Arnold Neumaier. Image Reconstruction Through Regularization by Envelope Guided Conjugate Gradients. In *Bell Labs Report 11274-94-14*. Submitted to IEEE Trans. Medical Imaging.

[11] John A. Scales. Iterative methods for large, sparse, inverse calculations. Samizdat Press. Colorado School of Mines, 76 Olcott Drive, White River Junction, Vermont 05001, April 1993. Lecture notes.

[12] S. R. Deans. *The Radon Transform and Some of Its Applications*. Krieger Publishing Company, Malabar, Florida, 2 edition, 1993. Previously John Wiley & Sons Inc., 1983.

[13] Per Christian Hansen. *Rank-Deficient and Discrete Ill-Posed Problems*. Polyteknisk Forlag, Lyngby, Denmark, 1996. Doctoral Dissertation.

[14] Peter Toft. *The Radon Transform - Theory and Implementation*. PhD thesis, Department of Mathematical Modelling, Technical University of Denmark, 1996.

[15] G. T. Herman and D. Odhner. Performance evaluation of an iterative image reconstruction algorithm for positron emission tomography. *IEEE Trans. Med. Imag.*, 10:336–246, 1991.

Appendix N

Mean Field Reconstruction with Snaky Edge Hints

This appendix contains the paper *Mean Field Reconstruction with Snaky Edge Hints*, by Peter Alshede Philipsen, Lars Kai Hansen and Peter Toft, which has been presented at the *Interdisciplinary Inversion Conference 95* in Aarhus. The paper will appear in *INVERSE METHODS - Interdisciplinary elements of Methodology, Computation and Application*, which will be published from Springer Publications in 1996 [11].

The paper has also been presented and published in the proceedings of the *Fourth Danish Conference for Pattern Recognition and Image Analysis 95* [12].

In the paper mean field techniques have been used to improve image quality by using strong priors in the restoration of PET reconstructed images. This work is based on the results shown in [61].

Mean Field Reconstruction with Snaky Edge Hints

Peter Alshede Philipsen, Lars Kai Hansen and Peter Toft
CONNECT, Electronics Institute, build. 349
Technical University of Denmark,
DK-2800 Lyngby, Denmark
email: pap,lkhansen,ptoft@ei.dtu.dk

16/6-95

1 Introduction

Reconstruction of imagery of a non-ideal imaging system is a fundamental aim of computer vision. Geman and Geman [4] introduced Metropolis sampling from Gibbs distributions as a simulation tool for visual reconstruction and showed that a *Simulated Annealing* strategy could improve the efficiency of the sampling process. The sampling process implements a stochastic neural network with symmetric connections. Peterson and Anderson applied the *Mean Field* approximation, and observed substantial improvements in speed and performance [6].

In the next section the Bayesian approach to reconstruction is outlined, as model example we study the so-called *Weak Membrane* model. The Weak Membrane is a popular vehicle for piece-wise smooth reconstruction and involves edge units (called line processes in [4]). Edge unit control has shown to be a major challenge in applications of the Weak Membrane. However, recently there has been substantial progress in use of deformable models for contour (2D) and surface (3D) modeling, see, e.g., [2, 3]. In this contribution we suggest to combine the two approaches, in particular we show how a "Snake" contour model may be used to produce efficient edge hints to the Weak Membrane. Section three contains experiments and concluding remarks.

2 Bayesian Visual Reconstruction

The basic idea is to consider both the source (un-degraded) signal and the processes of the imaging system as stochastic processes. The Bayes formula can then be used to obtain the distribution $P(V|d)$ of the reconstructed signal V, conditioned on the observed degraded signal d:

$$P(V|d) = \frac{P(d|V)P(V)}{P(d)} \quad (1)$$

This conditional distribution is the product of the distribution of the imaging system process: $P(d|V) \equiv P(V \rightarrow d)$, and the *prior* distribution of the reconstructed signal $P(V)$. $P(V|d)$ of Equation (1) is referred to as the *posterior* distribution. A useful estimate of the reconstructed signal is given by the location of the mode of the posterior distribution, the so-called *Maximum A Posteori* estimate.

2.1 The Weak Membrane Model

The *prior* distribution reflects our general insight on the objects being imaged, for example expressing that the image represents extended 3D structures etc. The Weak Membrane is a simple model for reconstruction of piece-wise continuous intensity surfaces (2D signals) from noisy observations [4]. In the lattice $(m,n) \in Z^2$ version the reconstructed surface is described by the intensity values $V_{m,n}$. The prior information formalizes the expectation that neighbor intensity values should be close, except when they are disconnected by the rare occasion of an active edge-unit. We introduce horizontal $h_{m,n}$ and vertical edges $v_{m,n}$. The *prior* probability distribution for a membrane reads:

$$P(V,h,v) = Z_1^{-1} \exp\left(-E_{\text{prior}}(y,h,v)\right) \tag{2}$$

where $E_{\text{prior}}(y,h,v)$ is the energy or cost function:

$$\begin{aligned} E_{\text{prior}}(V,h,v) &= \frac{\nu}{2}\left[\sum_{m,n}(1-h_{m,n})(V_{m,n}-V_{m+1,n})^2 + (1-v_{m,n})(V_{m,n}-V_{m,n+1})^2\right] \\ &+ \sum_{m,n}\mu^h_{m,n}h_{m,n} + \sum_{m,n}\mu^v_{m,n}v_{m,n} - \sum_{m,n}\gamma_{m,n}h_{m,n}h_{m,n+1}v_{m,n}v_{m+1,n} \end{aligned} \tag{3}$$

and Z_1^{-1} is a normalization constant. This is a *Gibbs* distribution[1]. Note that last terms in the cost function act as local *chemical potentials* for control of the number of active edge-units (i.e., $h_{m,n} = +1$).

The degradation process produces the measurements $(V_{m,n} \to d_{m,n})$. In the Weak Membrane example we will for simplicity study addition of zero-mean, white Gaussian noise with variance σ^2. Note that any other model of the degeneration process can be incorporated here:

$$P[d|V] = Z_2^{-1} \exp\left(-\frac{1}{2\sigma^2}\sum_{m,n}(V_{m,n}-d_{m,n})^2\right) \tag{4}$$

Using Bayes formula (1), we can combine Equations (2) and (4), to obtain the parameterized posterior distribution:

$$P[V,h,v|d] = Z^{-1}\exp(-E(V,h,v,d)) \tag{5}$$

where Z is a normalization constant and the energy function is given by,

$$E(V,h,v,d) = \frac{1}{2\sigma^2}\sum_{m,n}(V_{m,n}-d_{m,n})^2 + E_{\text{prior}}(V,h,v,d). \tag{6}$$

2.2 Network design

Inspecting the posterior distribution we note that the three-component process $(V_{m,n}, h_{m,n}, v_{m,n})$ realizes a *Compound Random Markov Field.* This property ensures that the neural network implementation only involves neighbor connectivity. To enhance the sampling efficiency we use a simulated annealing strategy as recommended by Geman and Geman [4]. The scheme is implemented by introducing a *temperature* T in the Gibbs distribution and design the sampler with a

[1] A distribution of the form $P(x) = Z^{-1}\exp(-E(x)/T)$, where $E(x)$ is a cost-function, bounded from below, and T is a parameter.

decreasing sequence of temperatures ending up at $T = 1$. The temperature dependent distribution reads:

$$P[V,h,v|d] = Z_T^{-1} \exp\left(-\frac{E[V,h,v,d]}{T}\right) \tag{7}$$

Sampling of this distribution, as investigated by Geman and Geman, is too slow for most applications and it is therefore recommended to invoke a deterministic approximation scheme to obtain the necessary averages [1, 6]. The self-consistent Mean Field equations for the Weak Membrane model in the k'th iteration read (see also [5] for an introduction):

$$\begin{aligned} V_{m,n}^{k+1} &= \kappa^{-1}\Big[\sigma^{-2}\, d_{m,n} + (1-h_{m,n}^k)V_{m+1,n}^k + (1-h_{m-1,n}^k)V_{m-1,n}^k \\ &\quad + (1-v_{m,n}^k)V_{m,n+1}^k + (1-v_{m,n-1}^k)V_{m,n-1}^k\Big] \\ h_{m,n}^{k+1} &= \tanh\left[\beta\left(\frac{1}{2}(V_{m,n}^k - V_{m+1,n}^k)^2 - \mu_h - \gamma_{m,n}\,\Gamma_{m,n}^h(k)\right)\right] \\ v_{m,n}^{k+1} &= \tanh\left[\beta\left(\frac{1}{2}(V_{m,n}^k - V_{m,n+1}^k)^2 - \mu_h - \gamma_{m,n}\,\Gamma_{m,n}^v(k)\right)\right] \end{aligned} \tag{8}$$

where $\beta = 1/T$ and

$$\Gamma_{m,n}^h(k) = h_{m,n+1}^k v_{m,n}^k v_{m+1,n}^k + h_{m,n-1}^k v_{m,n-1}^k v_{m+1,n-1}^k \tag{9}$$

$$\Gamma_{m,n}^v(k) = v_{m+1,n}^k h_{m,n}^k h_{m,n+1}^k + v_{m-1,n}^k h_{m-1,n}^k h_{m-1,n+1}^k \tag{10}$$

$$\kappa = \sigma^{-2} + 4 - h_{m,n}^k - h_{m-1,n}^k - v_{m,n}^k - v_{m,n-1}^k \tag{11}$$

We have used the same symbols for the averaged quantities as for the stochastic, but that should not lead to confusion since the two never occur in the same expression. The coupled equations are solved by straightforward iteration as shown in Equation (8), defining a recursive nearest neighbor connected *cellular neural network.*

2.3 Snake hints

There has been much recent progress in the use of deformable models for edge and surface identification, see, e.g., [2]. In this presentation we suggest to produce strong edge hints for the Weak Membrane through a Snake deformable model. The Snake is defined to be a periodic set of N points in the visual field of an image: $\mathbf{r}_j = (x_j, y_j)$. Periodicity meaning $\mathbf{r}_{j+pN} = \mathbf{r}_j, p \in Z$. The Snake energy function consists in a form control part E_{form} and a match part E_{match}.

$$E = E_{\text{form}} + E_{\text{match}} \tag{12}$$

In this simple version the form control preserves total length,

$$E_{\text{form}} = \frac{a}{2}(d - d_0)^2 \tag{13}$$

$$d = \sum_j |\mathbf{r}_j - \mathbf{r}_{j+1}| \tag{14}$$

where d_0 is the initial Snake length. More complex form controls can be designed that preserve shape, corners, etc. The Snake match energy is designed to ensure that the Snake points track edge contours in the image field. This effect may be obtained by letting the Snake points seek local maxima in the gradient energy map of the image, $G(\mathbf{r})$. The total Snake match energy is then given by,

$$E_{\text{match}} = -\sum_j G(\mathbf{r}_j). \tag{15}$$

Snake dynamics is established through gradient descent,

$$\frac{1}{\eta}\frac{\partial \mathbf{r}_j}{\partial t} = -\frac{\partial E}{\partial \mathbf{r}_j} = a(d-d_0)\frac{\partial d}{\partial \mathbf{r}_j} + \frac{\partial G(\mathbf{r}_j)}{\partial \mathbf{r}_j} \tag{16}$$

3 Experiments and concluding remarks

In numerous image processing applications quite detailed prior information can be devised. In, e.g., brain scan reconstruction detailed atlases are known describing the generic brain topography under various scanning modes. To illustrate the potential in brain scans of using strong edge hints we want to reconstruct a head phantom shown in Figure 1. This is done from a noise corrupted image shown in Figure 2. Under the reconstruction process we also want to estimate edges.

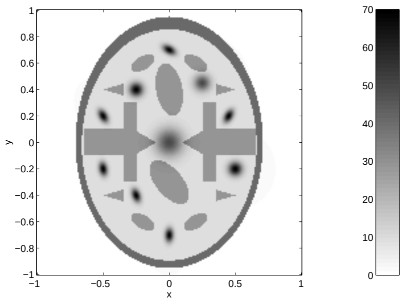

Figure 1: Original head phantom without noise.

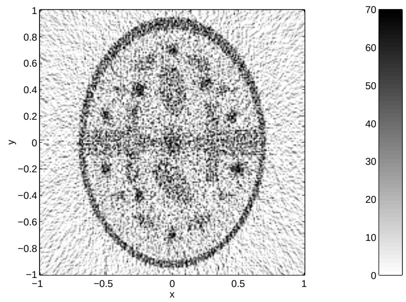

Figure 2: Noise corrupted head phantom shown in same color scale.

Snakes are initialized in seven generic positions, shown in Figure 3. For the particular instance the Snake equilibrates in about 100 iterations as illustrated in Figure 4. The approach to equilibrium is quite sensitive to the topography of the image and careful control of η is necessary. This problem may be relieved by use of a pseudo-second order search direction instead of the gradient of the edge energy map.

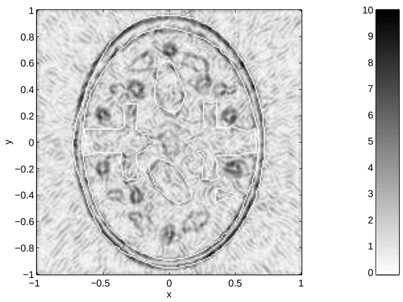

Figure 3: Edge energy map with initial snake positions shown in white.

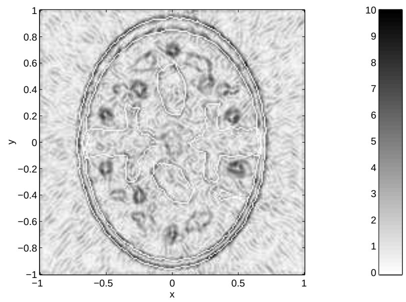

Figure 4: Edge energy map with equilibrated Snake positions shown in white.

Subsequently, hints are created by modulating the chemical potentials $\mu^h_{m,n}, \mu^v_{m,n}, \gamma_{m,n}$. This modulation imply that edge units are strongly suppressed outside of the region suggested by the Snakes. Finally we relax the cellular network as defined in Equation (8).

The result of the cellular net is presented in Figures 5 (without Snakes) and 6 (using Snakes). Note that the intensity level inside the region of the Snake has been estimated more closely using Snakes. This might be of significant importance in, e.g., brain activation studies, in which the minute differences in brain activity between activated and resting states are investigated. Using Snakes enhances the Signal to Noise ratio (SNR) with 12.5 dB. Without Snakes the improvement is 11.3 dB. The improvement is due to a better estimation of edges.

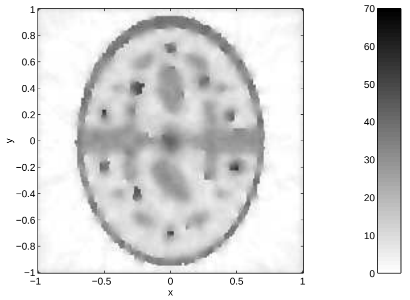

Figure 5: Restored head phantom using Mean Field Annealing.

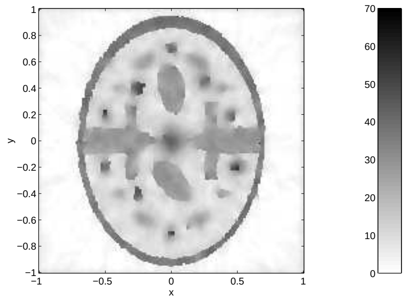

Figure 6: Restored head phantom using Snakes as edge priors to the Mean Field Annealing.

In Figures 7 and 8 are shown the active edges without using Snakes respectively with Snakes. As seen from Figure 8 it is possible to incorporate strong priors in order to obtain closed edge contours. This can be used to segment the image.

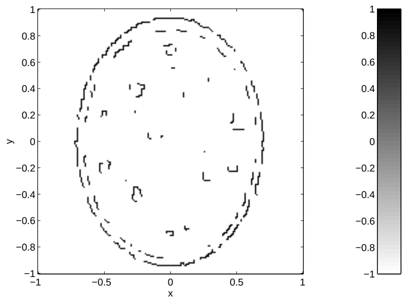

Figure 7: Estimated edges without using snakes. Active edges are marked black.

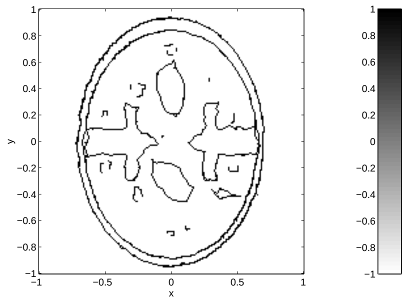

Figure 8: Estimated edges using Snakes as priors

Another way of evaluating the results is to examine the histograms of the images. Figure 9 shows the histogram of the original noise free image and Figure 10 shows, in the same interval, the histogram of the noise corrupted image.

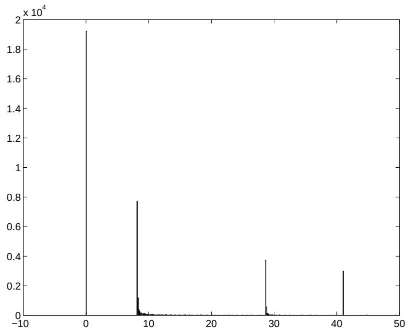

Figure 9: Histogram of original noise free head phantom.

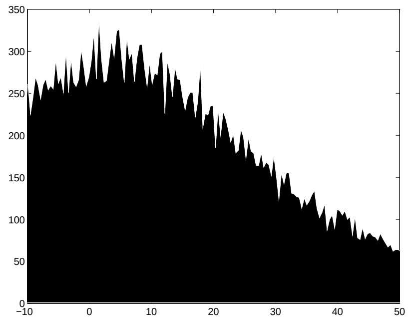

Figure 10: Histogram of the observed noise corrupted head phantom.

Without using Snakes the histogram after restoration can be seen in Figure 11. Finally Figure 12 shows the histogram corresponding the restored image using Snakes as a prior. From the Figures it can be seen that the two higher levels (29 and 41) is better resolved using Snakes and compared to the noisy histogram shown in Figure 10 the result is far better.

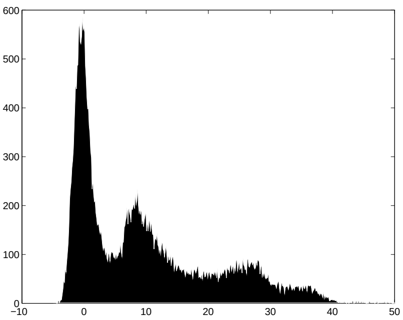

Figure 11: Histogram of the restored image without using Snakes.

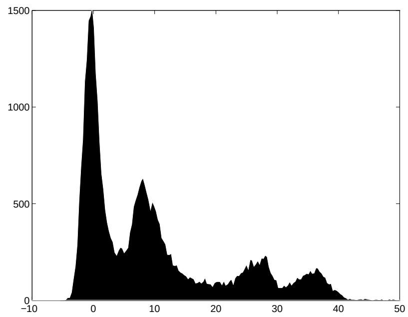

Figure 12: Histogram of the restored image using Snakes as priors.

In conclusion we have shown that strong structural priors may be introduced in the Weak Membrane model by invoking deformable models like Snakes. The present study was based on a head phantom; we are currently pursuing the viability of the approach in the context of Positron-Emission-Tomography.

Acknowledgment

This research is supported by the Danish Research Councils for the Natural and Technical Sciences through the Danish Computational Neural Network Center.

References

[1] A. Blake: *Comparison of the Efficiency of Deterministic and Stochastic Algorithms for Visual Reconstruction.* IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-11** 2-12 (1989).

[2] L. Cohen and I. Cohen: *Finite Element Methods for Active Contour Models and Balloons for 2-D and 3-D Images.* IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-15** 1133-1147 (1993).

[3] X. Ouyang, W.H. Wong, V.E. Johnson, X. Hu, C.-T. Chen: *Incorporation of Correlated Structural Images in PET Image Reconstruction.* IEEE Transactions on Medical Imaging **13** 627-640 (1994).

[4] D. Geman and S. Geman: *Stochastic Relaxation, Gibbs distributions and the Bayesian restoration of images.* IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-6** 721-741 (1984).

[5] J. Hertz, A. Krogh and R.G. Palmer: *Introduction to the Theory of Neural Computation.* Addison Wesley, New York (1991).

[6] C. Peterson and J.R. Anderson: *A Mean Field Learning Algorithm for Neural Networks.* Complex Systems **1** 995-1019, (1987).

[7] M.W. Roth: *Survey of Neural Network Technology for Automatic Target Recognition.* IEEE Transactions on Neural Networks **1**, 28-43, (1990).

Appendix O

Detection of Lines with Wiggles using the Radon Transform

This appendix contains the paper *Detection of Lines with Wiggles using the Radon Transform*, by Peter Toft [4].

The paper has been submitted to the *NORSIG'96 - 1996 IEEE Nordic Signal Processing Symposium* in Espoo, Finland.

Detection of Lines with Wiggles using the Radon Transform

Peter Toft

Department of Mathematical Modelling, Technical University of Denmark,
DK-2800 Lyngby, Denmark, Email pto@imm.dtu.dk.

Abstract

The discrete Radon transform is a useful tool in image processing for detection of lines (or in general curves) in digital images.

One of the key properties of the discrete Radon transform is that a line in an image is transformed into a peak in the parameter domain, where the position of the peak corresponds to the line parameters.

What often is needed, is to determine whether a Radon transform based curve detection algorithm will work in presence of noise. This paper regards detection of lines in images, where the lines are assumed to have wiggles. A theoretical analysis is given providing analytical expressions for this kind of noise.

I. Introduction

The Radon transform can be defined in various ways. In general image processing the lines are often parameterized using normal parameters [1], and in seismic signal processing the τ-p transform or slant stack version of the Radon transform use slope p and offset τ to parameterize the lines [2, 3]. Here the last definition will be used.

$$\check{g}(p,\tau) = \int_{-\infty}^{\infty} g(x, px+\tau)\, dx \tag{1}$$

One of the key properties of the (discrete) Radon transform is that a line in an image is transformed into a peak in the parameter domain [4], where the position of the peak corresponds to the line parameters. In this way, the Radon transform converts a difficult global detection problem in the image domain into a more easily solved local peak detection problem in the parameter domain.

For a discrete image $g(m,n)$, a nearest neighbour approximation of Eq. 1 will be used

$$\check{g}(\alpha,\beta) = \sum_{m=0}^{M-1} g\left(m, [\alpha m+\beta]\right) \tag{2}$$

where $[\cdot]$ denotes rounding to nearest neighbour. Lately analytical expressions for the probability of detecting a curve in presence of additive noise has been analyzed [5], and here another kind of noise is considered, namely that the lines in the images might not be perfectly linear but include some random misalignment, here called wiggles.

II. Lines with Wiggles

Assuming that a line in the image can be modelled as

$$g(m,n) = \delta(n - [\alpha^* m + \beta^* + \lambda]) \tag{3}$$

where $\delta(\cdot)$ is the Kronecker delta function, i.e., the Gaussian distributed noise term $\lambda \in \mathcal{N}(0,\sigma^2)$ determines the change in position of the line in the n-direction, and it will be assumed that the noise terms λ are uncorrelated as a function of m.

What now is considered, is to find the average value (and shape) of the peak (if any is found) in the discrete parameter domain as a function of the noise deviation σ. Eq. 3 implies that the probability of the sample $g(m,n)$ being 1 is given by

$$\begin{aligned} & P\{g(m,n)=1\} && (4)\\ =\; & P\{n = [\alpha^* m + \beta^* + \lambda]\} && (5)\\ =\; & P\left\{-\frac{1}{2} < n - \alpha^* m - \beta^* - \lambda < \frac{1}{2}\right\} && (6)\end{aligned}$$

and assuming that the sample point (m,n) is given from the nearest neighbour mapping in the discrete Radon transform, $n = [\alpha m + \beta]$, then the rounding function is modelled as an additive (noise) term ω

$$\begin{aligned} & P\{g(m,n)=1\} && (7)\\ =\; & P\left\{-\frac{1}{2} < [\alpha m+\beta] - \alpha^* m - \beta^* - \lambda < \frac{1}{2}\right\} && (8)\\ =\; & P\left\{-\frac{1}{2} < \alpha m+\beta+\omega - \alpha^* m - \beta^* - \lambda < \frac{1}{2}\right\} && (9)\\ =\; & P\left\{-\frac{1}{2} < \zeta+\omega-\lambda < \frac{1}{2}\right\} && (10)\\ =\; & \Phi\left(\frac{\frac{1}{2}+\zeta+\omega}{\sigma}\right) - \Phi\left(\frac{-\frac{1}{2}+\zeta+\omega}{\sigma}\right) && (11)\\ \zeta =\; & (\alpha-\alpha^*)m+\beta-\beta^* && (12)\end{aligned}$$

where $\Phi(\cdot)$ is the Gaussian probability function, and ζ is a displacement between the true line parameters (α^*, β^*) and the parameters (α, β) at a given position m.

Due to many values of m used in Eq. 2, it is a reasonable approximation to model ω as a uniformly distributed variable between the limits $-1/2$ and $1/2$, i.e., $\omega_m \in \mathcal{U}(-1/2, 1/2)$. Eq. 11 implies that the average contribution from the pixel $g(m,n)$ to the discrete Radon transform is given by

$$E\{\breve{g}(\alpha,\beta)|g(m,n)\} = \int_{\omega=-\frac{1}{2}}^{\omega=\frac{1}{2}} P_m(\omega)\; d\omega \tag{13}$$

$$= \int_{\omega=-\frac{1}{2}}^{\omega=\frac{1}{2}} \Phi\left(\frac{\frac{1}{2}+\zeta+\omega}{\sigma}\right) - \Phi\left(\frac{-\frac{1}{2}+\zeta+\omega}{\sigma}\right)\; d\omega \tag{14}$$

In order to get an analytical expression for $E\{\breve{g}(\alpha,\beta)|g(m,n)\}$ the Gaussian probability function is approximated

$$\Phi(x) \approx \frac{1}{2}\left(1+\tanh\left(\frac{x}{\gamma}\right)\right) \quad \text{where } \gamma = \sqrt{\frac{\pi}{2}} \tag{15}$$

i.e., the integral of $\Phi(\cdot)$ can be approximated by

$$\int \Phi(x)\; dx \approx \frac{\gamma}{2}\left(\frac{1+x}{\gamma} + \log\cosh\left(\frac{x}{\gamma}\right)\right) \tag{16}$$

where $\log(\cdot)$ is the natural logarithm.

If Eq. 16 is inserted into Eq. 14, followed by some rearrangements of the expressions, it is found that

$$E\{\breve{g}(\alpha,\beta)|g(m,n)\} = \frac{\gamma\sigma}{2}\log\left(\frac{\cosh\left(\frac{2\zeta}{\gamma\sigma}\right)+\cosh\left(\frac{2}{\gamma\sigma}\right)}{\cosh\left(\frac{2\zeta}{\gamma\sigma}\right)+1}\right) \tag{17}$$

In Fig. 1 the average weight to the discrete Radon transform is shown as a function of σ and ζ.

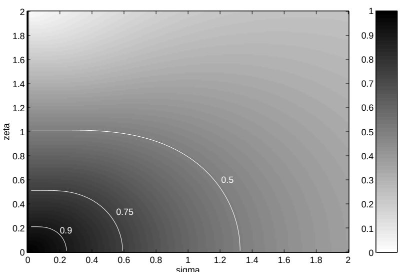

Figure 1 The average weight to the discrete Radon transform $E\{\breve{g}(\alpha,\beta)|g(m,n)\}$ as a function of the noise deviation σ on the first axis, and the displacement ζ on the second axis.

A special case of interest regards $\zeta = 0$, i.e., where the coordinates of the line in the image matches exactly a sample in the discrete parameter domain.

$$\zeta = 0 \;\Rightarrow\; E\{\breve{g}(\alpha,\beta)|g(m,n)\} \tag{18}$$

$$= \sigma\sqrt{\frac{\pi}{8}}\log\left(\frac{1}{2}\left(1+\cosh\left(\frac{\sqrt{8}}{\sigma\sqrt{\pi}}\right)\right)\right) \tag{19}$$

In Fig. 2 the average weight to the discrete Radon transform is shown as a function of σ, when the displacement is assumed negligible. It can be seen that the function agrees well with a simulated result shown in Fig. 3, that is found by generating 10000 images with the correct noise amplitude σ and computing the discrete Radon transform only for the parameter set matching the true parameters. An even better agreement can be found using a numerical integration of Eq. 14, but Eq. 16 provides an simple analytical result that approximates the simulated results well, though it can also be seen that the theoretical model has a small bias, when $\sigma \approx 0.1$.

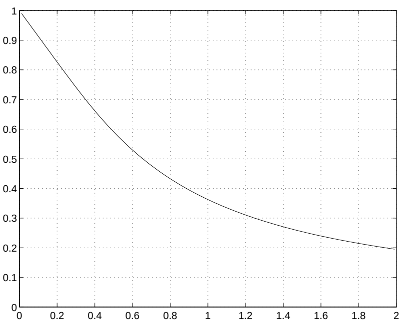

Figure 2 The theoretical result of the average weight to the discrete Radon transform, found in Eq. 19, as a function of σ when the displacement ζ is negligible.

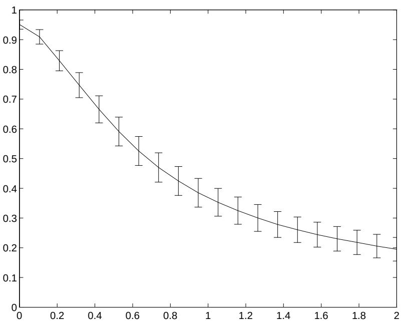

Figure 3 The simulated average value of the peak using 10000 images for each noise level. The vertical error bars show the measured deviation.

Fig. 2 or Eq. 19 can be used to predict whether line parameters might be estimated given that the lines have wiggles. If, e.g., demanding that the peak value cannot decrease more than 50% due to wiggles, then σ cannot exceeds approximately 0.7. And a 25% decrease of the peak value is found if $\sigma \approx 0.25$.

III. Examples

In Fig. 4 is shown an image containing one line without noise, and Fig. 5 shows the discrete parameter domain in the area of the peak. Next, Fig. 6 shows an image where $\sigma = 0.5$, and Fig. 7 shows the corresponding discrete parameter domain. On the figure-axes, the continuous variables, indicated in Eq. 1, have been used. This merely imply a linear scaling of the discrete parameters used in Section II.. The images have been scaled individually according to the minimal and maximal values, and it can be seen that the maximal value here is around 65, and the shape is more uneven than seen from Fig. 5. This value lies within the error bars shown in Fig. 3.

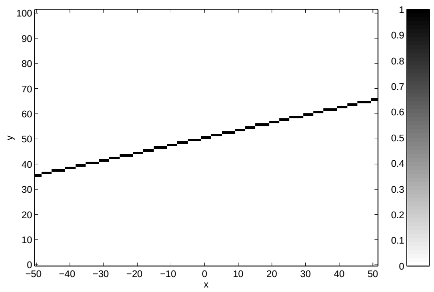

Figure 4 An image containing one line without any kind of noise.

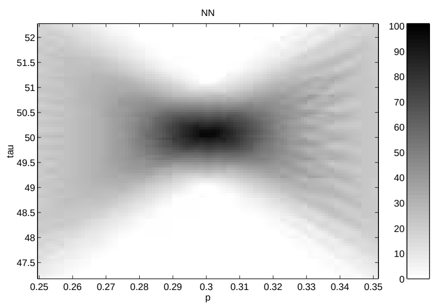

Figure 5 The discrete Radon transform of the image shown in Fig. 4 zoomed in at the peak.

IV. Conclusion

The conclusion is that a peak is generated in the discrete parameter domain corresponding to each of the wiggly line in the image, and the value of the peak will be lowered corresponding to the noise level. Analytical expressions have been given to determine the actual reduction of the peak value as a function of the noise level.

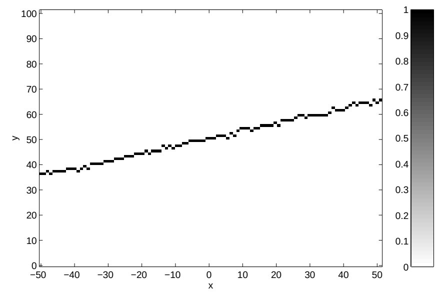

Figure 6 An image containing one wiggly line where $\sigma = 0.5$.

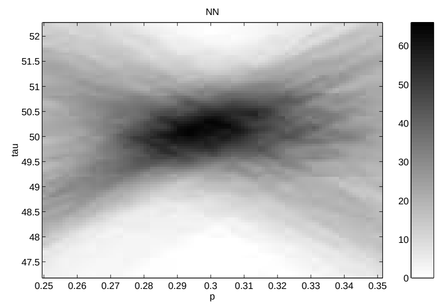

Figure 7 The discrete Radon transform of the image shown in Fig. 6 zoomed in at the peak.

References

[1] S. R. Deans. *The Radon Transform and Some of Its Applications.* Krieger Publishing Company, Malabar, Florida, 2 edition, 1993. Previously John Wiley & Sons Inc., 1983.

[2] Gregory Beylkin. Discrete Radon Transform. *IEEE Tr. ASSP*, 35(2):162–172, 1987.

[3] T. S. Durrani and D. Bisset. The Radon Transform and its Properties. *Geophysics*, 49(8):1180–1187, Aug. 1984. Correction note in vol. 50, no. 5, pp. 884–886, May 1985.

[4] Peter Toft. *The Radon Transform - Theory and Implementation.* PhD thesis, Department of Mathematical Modelling, Technical University of Denmark, 1996.

[5] Peter Toft. Using the Generalized Radon Transform for Detection of Curves in Noisy Images. In *Proceedings*, volume 4, pages 2221–2225. IEEE ICASSP, May 1996.

Thesis Bibliography

[1] K. V. Hansen and P. A. Toft. Fast Curve Estimation Using Pre-Conditioned Generalized Radon Transform. *IEEE Image Processing*, 1996. To appear in IEEE Image Processing. Preprint Department of MAthematical Modelling, Technical University of Denmark, DK2800 Lyngby.

[2] P. A. Toft and K. V. Hansen. Fast Radon Transform for Detection of Seismic Reflections. In *Signal Processing VII - Theories and Applications*, volume I, pages 229–232. EURASIP EUSIPCO94, 1994.

[3] Peter Toft. Using the Generalized Radon Transform for Detection of Curves in Noisy Images. In *Proceedings*, volume 4, pages 2221–2225. IEEE ICASSP, May 1996.

[4] Peter Toft. Detection of Lines with Wiggles using the Radon Transform. *Submitted to NORSIG'96*, 1996. Preprint from Department of Mathematical Modelling, Technical University of Denmark.

[5] Peter Toft and Jesper James Jensen. A Very Fast Implementation of 2D Iterative Reconstruction Algorithms. *Submitted IEEE Medical Imaging Conference*, 1996.

[6] Peter Toft and Jesper James Jensen. Accelerated 2D Iterative Reconstruction. *Submitted*, 1996. Preprint from Department of Mathematical Modelling, Technical University of Denmark.

[7] Søren Holm Peter Toft and Mikael Jensen. Estimation of the noise contributions from Blank, Transmission and Emission scans in PET. In *To appear in proceedings*. IEEE MIC, October 1995.

[8] Søren Holm, Peter Toft, and Mikael Jensen. Estimation of the noise contributions from Blank, Transmission and Emission scans in PET. *IEEE Transactions on Nuclear Science*, 1996. Submitted.

[9] Technical University of Denmark. Section for Digital Signal Processing Department of mathematical Modelling. Homepage of the danish part of the Human Brain Project. `http://hendrix.ei.dtu.dk`.

[10] Peter Toft. Homepage of Peter Toft. Section for Digital Signal Processing IMM. `http://www.ei.dtu.dk/staff/ptoft/ptoft.html`.

[11] Peter Alshede Philipsen, Lars Kai Hansen and Peter Toft. Mean Field Reconstruction with Snaky Edge Hints. In *To appear in INVERSE METHODS - Interdisciplinary elements of Methodology, Computation and Application*, 1995.

[12] Peter Alshede Philipsen, Lars Kai Hansen and Peter Toft. Mean Field Reconstruction with Snaky Edge Hints. In Peter Johansen, editor, *Proceedings fra den Fjerde Danske Konference om Mønstergenkendelse og Billedanalyse*. DIKU, August 1995.

[13] J. Radon. Über die bestimmung von funktionen durch ihre integralwerte längs gewisser mannigfaltigkeiten. *Ber. Ver. Sächs. Akad. Wiss. Leipzig, Math-Phys. Kl.*, 69:262–277, April 1917. In German. An english translation can be found in S. R. Deans: *The Radon Transform and Some of Its Applications*, app. A.

[14] S. R. Deans. *The Radon Transform and Some of Its Applications*. Krieger Publishing Company, Malabar, Florida, 2 edition, 1993. Previously John Wiley & Sons Inc., 1983.

[15] S. R. Deans. Hough Transform From the Radon Transform. *IEEE PAMI*, 3(2):185–188, 1981.

[16] Peter Toft. Radontransformation. Lecture Note for DTU course 4235: Advanced Digital Signal Processing, Electronics Institute, Technical University of Denmark, 1994. In Danish.

[17] J. Canny. A Computational Approach to Edge Detection. *IEEE Trans. PAMI*, 8(6):679–698, 1986.

[18] R. Deriche. Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector. *International Journal of Computer Vision*, 2(1):167–187, 1987.

[19] Özdoğan Yilmaz. *Seismic Data Processing*. Society of Exploration Geophysicists, Tulsa, Oklahoma, 1987.

[20] T. S. Durrani and D. Bisset. The Radon Transform and its Properties. *Geophysics*, 49(8):1180–1187, Aug. 1984. Correction note in vol. 50, no. 5, pp. 884–886, May 1985.

[21] Athanasios Papoulis. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, 2. edition, 1984.

[22] Peter Toft. RadonAna: A C-based Package for Analytically Based Radon Transformation. Technical report, Electronics Institute, DTU, 1995.

[23] P. V. C. Hough. A Method and Means for Recognizing Complex Patterns. US Patent: 3,069,654, Dec. 1962.

[24] R. O. Duda and P. E. Hart. Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Communication of the Association for Computing Machinery*, 15(1):11–15, 1972.

[25] J. Illingworth and J. Kittler. A Survey of the Hough Transform. *Computer Vision, Graphics, and Image Processing*, 44:87–116, 1988.

[26] V. F. Leavers. *Shape Detection in Computer Vision Using the Hough Transform*. Springer-Verlag, 1992.

[27] Wilson Lam et al. An Analysis on Quantizating the Hough space. *Pattern Recognition Letters*, 15:1127–1135, Nov. 1994.

[28] J. Princen et al. A Comparison of Hough Transform Methods. In *Third International Conference on Image Processing and its Applications*, pages 73–77. IEE, 18-20 July 1989.

[29] J. Princen et al. A Formal Definition of the Hough Transform: properties and Relationships. *Journal of Mathematical Imaging and Vision*, 1:153–168, 1992.

[30] Lei Xu et al. A new curve detection method: Randomized Hough Transform (RHT). *Pattern Recognition Letters*, 11(5):331–338, 1990.

[31] Keikke Kälviäinen et al. Probabilistic and Non-probalistic Hough Transforms: Overview and Comparations. In *Randomized Hough Transform: New Extentions*, number 35 in Research Letters. Lappeenranta University of Technology, 1994. email: Heikki.Kalvianen@lut.fi.

[32] Katja Blankensteiner og Peter Toft. Development and Implementation of a Fast and Robust Algorithm for Estimation of the Generalized Radon Transform for Seismical Signal Processing. Master's thesis, Electronics Institute. Technical University of Denmark, 1993. In danish.

[33] Piang Liang. A New and Efficient Transform for Curve Detection. *Journal of Robotic Systems*, 8(6):841–847, 1991.

[34] D. H. Ballard. Generalizing the Hough Transform to detect arbitrary shapes. *Pattern Recognition Letters*, 13(2):111–122, 1981.

[35] H. Li, M. A. Lavin and R. J. Le Master. Fast Hough Transform: A Hierachical Approach. *Computer Vision, Graphics, and Image Processing*, 36:139–161, 1986.

[36] Gregory Beylkin. Discrete Radon Transform. *IEEE Tr. ASSP*, 35(2):162–172, 1987.

[37] C. J. Haneveld and G. C. Herman. A Fast Algorithm for Computation of Radon Transforms. *Geophysical Prospecting*, 38:853–860, 1990.

[38] W. A. Schneider. The Common Depth Point Stack. *Proceedings of the IEEE*, 72(10):1238–1254, 1984.

[39] E. A. Robinson, T. S. Durrani and L. G. Peardon. *Geophysical Signal Processing*. Englewood Cliffs, New Jersey, Prentice-Hall Inc., 1980.

[40] B. S. Atal and J. R. Remde. A New Model of LPC Excitation for Producing Natural-Sounding Speech at Low Bit Rates. In *Proceedings of ICASSP*, volume 1 of *ICASSP*, pages 614–617. IEEE, 1982.

[41] M. Cookey, H. J. Trussel and I. J. Won. Seismic Deconvolution by Multipulse Methods. *IEEE Trans. ASSP*, 38(1):156–160, 1990.

[42] Kim Vejlby Hansen. *Detection of Primary Reflection Curves in CMP gathers*. PhD thesis, Electronics Institute, Technical University of Denmark, Joint work with Ødegaard & Danneskiold-Samsøe., 1994.

[43] George A. McMechan and Richard Ottolini. Direct observation of a p-τ curve in a slant stacked wave field. *Bulletin of the Seismological Society of America*, 70(3):775–789, 1980.

[44] Robert Phinney et al. Transformation and Analysis of Record Sections. *Journal of Geophysical Research*, 86(B1):359–377, Jan. 1981.

[45] John B. Diebold and Paul L. Stoffa. The Traveltime Equation, tau-p mapping, and inversion of common midpoint data. *Geophysics*, 46(3):238–254, March 1981.

[46] Friedemann Wenzel et al. Seismic modelling in the domain of intercept time and ray parameter. In The Multidimensional Signal Processing Society, editor, *Selected Papers in Multidemensional Digital Signal Processing*, pages 377–393. IEEE Acoustics, Speech, and Signal Processing Society, 1986.

[47] Paul Stoffa et al. Direct mapping of seismic data to the domain of intercept time and ray parameter - a plan-wave decomposition. *Geophysics*, 46(3):255–267, March 1981.

[48] K. V. Hansen and J. Larsen. An Algorithm for Successive Identification of Reflections. *IEEE IP*, 3(3):281–291, May 1994.

[49] K. Y. Huang *et al.* Image Processing of Seismograms: (A) Hough Transformation for the Detection of Seismic Patterns; (B) Thinning Processing in the Seismogram. *Pattern Recognition*, 18(6):429–440, 1985.

[50] C. Y. Chi, J. M. Mendel and D. Hampson. A Computationally Fast Approach to Maximum-Likelihood Deconvolution. *Geophysics*, 49(5):550–565, 1984.

[51] A. Blake and A. Zisserman. Comparison of the Efficiency of Deterministic and Stochastic Algorithms for Visual Reconstruction. *IEEE Trans. Neural Networks*, 11(1):2–12, 1989.

[52] Michel Lavergne. *Seismic Methods.* Graham & Trotman Limited, Sterling House, London, 2 edition, 1989. Also at Krieger Inc.

[53] Bjørn Ursin. Seismic velocity estimation. *Geophysical Prospecting*, 25:658–666, 1977.

[54] C. Hewitt Dix. Seismic Velocities from Surface Measurements. *Geophysics*, XX(1):68–86, Jan. 1955.

[55] Enders A. Robinson and Eduard J. Douze. Ray tracing and seismic modelling. In *Proceedings of the Fourteenth International Symposium*, volume 14 of *Acoustical Imaging*, pages 169–186, 1985.

[56] Lesley M. Murphy. Linear feature detection and enhancement in noisy images via the Radon transform. *Pattern Recognition Letters*, 4:279–284, Sep. 1986.

[57] Department of Molecular and UCLA Medical Pharmacology. Homepage: Let's Play PET. `http://www.nuc.ucla.edu/lpp/lpphome.html`.

[58] J. Carlson et al. Fundamentals of medical imaging. Notes for the tutorial course with title Fundamentals of Medical Imaging held at the IEEE Medical Imaging Conference (MIC) 1995., 1995.

[59] Jesper J. Jensen and Peter A. Philipsen. Direkte Methods til rekonstruction af PET billeder. Master's thesis, Electronics Institute. Technical University of Denmark, 1995. In danish.

[60] Jesper J. Jensen. Iterative Methoder til rekonstruktion af PET billeder. Master's thesis, Electronics Institute. Technical University of Denmark, 1995. In danish.

[61] Peter A. Philipsen. Preprocessering og postprocessering af PET billeder. Master's thesis, Electronics Institute. Technical University of Denmark, 1995. In danish.

[62] Caroline Axelsson. *Direct Fourier Methods in 3D-Reconstruction from Cone-Beam Data.* PhD thesis, Linköbing University, Dep. of Electrical Engineering, S-581 83 Linköbing, Sweden, Jan. 1994. E-mail caroline@isy.liu.se.

[63] Frank Natterer. *The Mathematics of Computerized Tomography.* John Wiley & Sons, 2 edition, 1986.

[64] S. S. Orlov. Theory of three dimensional reconstruction. 1. Conditions for a complete set of projections. *Sov. Phys. Crystallogr.*, 20(3):312–314, 1975.

[65] S. S. Orlov. Theory of three dimensional reconstruction. The recovery operator. *Sov. Phys. Crystallogr.*, 20(4):429–433, 1975.

[66] James G. Colsher. Fully three-dimensional positron emission tomography. *Physics in Medicine and Biology*, 25(1):103–115, 1980.

[67] R. Clack. Towards a complete description of three-dimensional filtered backprojection. *Physics in Medicine & Biology*, 37(3):645–660, March 1992.

[68] R. Clack, D. Townsend and M. Defrise. An Algorithm for three-dimensional Reconstruction Incorporating Cross-Plane Rays. *IEEE Trans. Med. Imag.*, MI-8(1):32–42, 1989.

[69] Charles Stearns. *Accelerated Image Reconstruction For A Cylindrical Positron Tomograph Using Fourier Domain Methods.* PhD thesis, Massachusetts Institute of Technology, 1990.

[70] J. G. Rogers, R. Harrop and P. E. Kinahan. The Theory of Three-Dimensional Image Reconstruction for PET. *IEEE Trans. Med. Imag.*, MI-6(3):239–243, September 1987.

[71] P. E. Kinahan and J. G. Rogers. Analytic 3D Image Reconstruction Using All Detected Events. *IEEE Transactions on Nuclear Science*, 36(1):964–968, Feb. 1989.

[72] Maria Magnusson. *The Linogram Method for Image Reconstruction from Projections.* PhD thesis, Linköbings tekniska högskola, Dep. of Electrical Engineering, S-581 83 Linköbing, Sweden, 1889. Revised version 1993.

[73] A. K. Jain. *Fundamentals of Digital Image Processing.* Prentice Hall, Baltimore and London second, 2 edition, 1989.

[74] A. Mallon and P. Grangeat. Three-dimensional PET reconstruction with time-of-flight measurements. *Physics in Medicine & Biology*, 37(2):717–29, 1992.

[75] Peter Toft. Using the Inverse Radon transform in PET. Technical report, Technical University of Denmark, Electronics Institute, DK2800 Lyngby, 1994.

[76] Michael Lautsch. A Spline Inversion Formula for the Radon Transform. *SIAM J. Numer. Anal.*, 26(2):456–467, 1889.

[77] Yves Nievergelt. Elementary Inversion of Radon's Transform. *SIAM Review*, 28(1):79–84, March 1986.

[78] R. M. Mersereau. Recovering Multidimensional Signals from their Projections. *Computer Graphics and Image Processing*, 1(4):179–195, Dec. 1973.

[79] Dan E. Dudgeon and Russel M. Mersereau. *Multidimensional Digital Signal Processing.* Prentice-Hall, Englewood Cliffs,N.J, 1984.

[80] L. A. Shepp and J. B. Krustal. Computerized tomography: The new medical x-ray technology. *Am. Math. Monthly*, 85:420–439, April 1978.

[81] A. Rosenfeld and A. C. Kak. *Digital Picture Processing*, volume 1. Academic Press, 2. ed. edition, 1982.

[82] Sigurdur Helgason. *The Radon Transform*, volume 5 of *Progress in Mathematics*. Birkhäuser, Boston Basel Stuttgart, 1980.

[83] P. A. M. Dirac. *The Principles of Quantum Mechanics*. Oxford University Press, Oxford, 1984.

[84] Gregory Beylkin. The Inversion Problem and Application of the Generalized Radon Transform. *Communications on Pure and Applied Mathematics*, XXXVII:579–599, 1984.

[85] C. H. Chapman. Generalized Radon transforms and slant stacks. *Geophys. J. R. ast. Soc.*, 66:445–453, 1981.

[86] Lennart Råde and Bertil Westergren. *BETA: Mathematics Handbook*. Chartwell-Bratt Ltd, 1980.

[87] A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing*. Englewood Cliffs, New Jersey, Prentice-Hall Inc., 1989.

[88] Simon Boel Pedersen. *Signalanalyse III*. Elektronisk Institut, DTU, 1990. In danish.

[89] Peter Koefoed Møller. *Den hurtige Fouriertransformation (FFT)*. Electronics Institute, DTU, 1991. In Danish.

[90] William H. Press *et. al*. *Numerical Recipes in C*. Cambridge University Press, 2. edition, 1992.

[91] Paul Kinahan. Personal Communication.

[92] P. R. Edholm and G. T. Herman. Linograms in Image Reconstruction from Projections. *IEEE Trans. Med. Imag.*, 6(7):301–307, Dec. 1987.

[93] Kevin J. Moraski and David C. Munson, Jr. Fast tomographic reconstruction using chirp-z interpolation. *IEEE Trans. Med. Imag.*, 1991.

[94] L. R. Rabiner et al. The Chirp z-Transform Algorithm. *IEEE Trans. Audio and Electroacoustics*, AU-17(2):86–92, 1969.

[95] L. I. Bluestein. A linear filtering approach to the computation of the discrete Fourier transform. *NEREM Rec.*, pages 218–219, 1968.

[96] Yair Censor. Finite Series-Expansion Reconstruction Methods. In *Proc. of the IEEE*, volume 71, pages 409–419, March 1983.

[97] Julia K. Older and Paul C. Johns. Matrix formulation of computed tomography reconstruction. *Physics in Medicine & Biology*, 38:1051–1064, 1993.

[98] Robert M. Lewitt. Alternatives to voxels for image representation in iterative reconstruction algorithms. *Phys. Med. Biol.*, 37(2):706–716, 1992.

[99] Jules S. Jaffe. Limited Angle Reconstruction Using Stabilized Algorithms. *IEEE Trans. Med. Imag.*, 9(3):338–344, Sep. 1990.

[100] S. Kawata and O. Nalcioglu. Constrained iterative Reconstruction by the Conjugate Gradient Method. *IEEE Trans. Med. Imag.*, MI-4(2):65–71, June 1985.

[101] Per Christian Hansen. *Rank-Deficient and Discrete Ill-Posed Problems.* Polyteknisk Forlag, Lyngby, Denmark, 1996. Doctoral Dissertation.

[102] G. H. Golub and C. F. Van Loan. *Matrix Computations*, volume 4. The Johns Hopkins University Press, Baltimore and London second, 1991.

[103] Per Christian Hansen. Personal Communication.

[104] Mark F. Smith. Simultaneously constraining spect activity estimates with primary and secondary energy window projection data. *IEEE Trans. Med. Imag.*, 13(2), June 1994.

[105] Mark F. Smith. Generalized matrix inverse reconstruction for spect using a weighted singular value spectrum. In *Abstract Book of IEEE Nuclear Science Symposium and Medical Imaging Conference*, page 161. IEEE, 1995. Also see the Conference Issue.

[106] Gabor T. Herman and Lorraine B. Meyer. Algebraric Reconstruction Techniques Can Be Made Computationally Efficient. *IEEE Trans. Med. Imag.*, 12(3):600–609, Sep. 1993.

[107] S. Kaczmarz. Angenährte Auslösung von Systemen linearer Gleichungen. *Bull. Int. Acad. Pol. Sci. Lett. A*, 35:355–357, 1937.

[108] Huaiqun Guan and Richard Gordon. A projection access order for speedy convergence of ART (algebraric reconstruction technique): a multilevel scheme for cumputed tomography. *Phys. Med. Biol.*, 39:2005–2022, 1994.

[109] G. T. Herman. *Image Reconstruction from Projections: The fundamentals of Computerized Tomography.* Academic Press, 1980.

[110] Paul E. Kinahan and Joel Karp. Determining when 3D reconstruction methods are required in volume PET imaging. In *Conference Record*, volume 2, pages 904–906. IEEE Nuclear Science Symposium & Medical Imaging Conference, 1992.

[111] Linda Kaufman. Implementing and Accelerating the EM Algorithms for Positron Emission Tomography. *IEEE Trans. Med. Imag.*, MI-6(1):37–51, march 1987.

[112] L. A. Shepp Y. Vardi and L. Kaufman. A Statistical Model for Positron Emission Tomography. *Journal of the American Statistical Association*, 80(389):8–37, March 1985. The pages include comments and discussion by several authors.

[113] R. E. Carson and K. Lange. The EM parametric image reconstruction algorithm. *Journal of the American Statistical Association*, 389(80):20–25, March 1985.

[114] G. T. Herman and D. Odhner. Performance evaluation of an iterative image reconstruction algorithm for positron emission tomography. *IEEE Trans. Med. Imag.*, 10:336–246, 1991.

[115] Linda Kaufman. Maximum Likelihood, Least Squares, and Penalized Least Squares for PET. *IEEE Trans. Med. Imag.*, 12(2):200–214, June 1993.

[116] John M. Ollinger. Maximum-Likelihood reconstruction of transmission images in emission computed tomography via the EM algorithm. *IEEE Trans. Med. Imag.*, 13(1):89–101, March 1994.

[117] John A. Scales. Iterative methods for large, sparse, inverse calculations. Samizdat Press. Colorado School of Mines, 76 Olcott Drive, White River Junction, Vermont 05001, April 1993. Lecture notes.

[118] Linda Kaufman and Arnold Neumaier. Image Reconstruction Through Regularization by Envelope Guided Conjugate Gradients. Technical report, Bell Labs, 1993. Report 11274-94-14.

[119] Jesper James Jensen. *3D Visualisering*. IMM Technical University of Denmark, `ftp://ei/dist/1995/james.3Dvis.ps.Z` and ftp://ei/dist/Radon/polyr.tar.gz. Manual in Danish. A software package called Polyr for converting a volume into a surface mesh has also been made.

[120] Technical University of Denmark. Section for Digital Signal Processing. Home of the OOGL database of 3D objects. `http://hendrix.ei.dtu.dk/oogl/ooglhome.html`.

[121] M. Daube-Witherspoon and G. Muehllehner. Treatment of axial data in threee-dimensional PET. *J. Nucl. Med.*, 28:1717–1724, 1987.

[122] M. Defrise. A factorization method for the 3D x-ray transform. *Inverse Problems*, 11:983–994, 1995.

[123] The Geometry Center. Homepage of The Geometry Center. Authors of Geomview. `http://www.geom.umn.edu`.

[124] A. E. Todd-Pokropek and P. H. Jarrit. The noise characteristics of SPECT systems. In P. J. Ell and B. L. Holman, editor, *Computed Emission Tomography*, pages 361–389. Oxford University Press, Oxford, 1982.

[125] N. M. Alpert et al. Estimation of the Local Statistical Noise in Emission Computed Tomography. *IEEE Trans. Med. Imag.*, MI-1(2):142–146, Oct. 1982.

[126] C. W. Stearns and D. Wack. A Noise Equivalent Counts Approach to Transmission Imaging and Source Design. *IEEE Trans. Med. Imag.*, MI-12(2):287–292, June 1993.

[127] S. C. Strother et al. Measuring PET Scanner Sensitivity: Relating Count Rates to Image Signal-to-Noise Ratios Using Noise Equivalent Counts. *IEEE Trans. Nucl. Sci.*, NS-37(2):783–788, April 1990.

[128] T. R. DeGrado et al. Performance Characteristics of a Whole-Body PET Scanner. *J. Nucl. Med.*, 35(8):1398–1406, 1994.

[129] T. K. Lewellen et al. Investigations of the Count Rate Performance of General Electric Advance Positron Emission Tomograph. *IEEE Trans. Nucl. Sci.*, NS-42(4):1051–1057, August 1995.

[130] T. Lewellen et al. Investigations of the Performance of the General Electric Advance Position Emission Tomograph in 3D mode. In *Submitted*, 1995.

[131] P. S. Crandall and C. W. Stearns. A Scalable Multi Processor Implementation of the Reprojection Algorithm for Volumetric PET Imaging. In *Conference Record of the IEEE 1995 Nuclear Symposium and Medical Imaging Conference.*, 1995. In press.

[132] S. Holm et al. 3D PET activation studies with $H_2{}^{15}O$ bolus injection. Count rate performance and dose optimization. In *Quantification of Brain Function using PET.* Academic Press, San Diego, 1995. In press.

[133] S. Holm et al. Count Rate Performance and Dose Optimization for rCBF with O-15 Water Bolus Injection in the New 3D PET. *J. Nucl. Med.*, 36:105P, 1995.

[134] S. Pajavic et al. Noise Properties of 3D PET Images. *J. Nucl. Med.*, 36:p. 105 (abstract), 1995.

[135] S. R. Meikle et al. Attenuation Correction Using Count-Limited Transmission Data in Positron Emission Tomography. *J. Nucl. Med.*, 34:143–150, 1993.

[136] X. Ouyang *el al.* Incorporation of Correlated Structural Images in PET Image Reconstruction. *IEEE Trans. Med. Imag.*, 13:627–640, 1994.

[137] Chien-Min Kao et al. Improved Bayesian Image Reconstruction in Positron Emission Tomography. In *Conference Issue of IEEE Medical Imaging Conference*, 1995. To appear.

[138] L. Cohen and I. Cohen. Finite Element Methods for Active Contour Models and Balloons for 2-D and 3-D Images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1993.

[139] Brian Cabral, Nancy Cam, and Jim Foran. Accelerated Volume Rendering and Tomographic Reconstruction using Texture Mapping Hardware. In *1994 Symposium on Volume Visualization*, pages 91–98. ACM SIGGRAPH, ACM Press, oct 1994.

[140] Gel'fand et al. *Generalized Functions*, volume 5. Academic Press, Baltimore and London second, 2 edition, 1964.

Index