Tomographical experiement

Shorter english version of original report
and
Code insights

Mika Lindström
mika.lindstroem@gmail.com

Measures done: 20.10.2016
Original finnish report given to UEF archieve: 10.11.2016

Contents

Short note

In this work, which is given task to students, we find a way to get knowledge about an objects interior with radiation based tomography measurement. The object is to be radiated with a γ-radiator (photon-radiator). The radiation ray attentuates within object and we can calculate the attentuation distribution of the object and build an image of it. We use Tikhonov regularisation with different Tikhonov-matrices to enchant the image. Image matrices' condition values are also studied.

If you are about to repeat this experiement, please note that it is important to understand the image reconstructive algorithm by yourself. Best practise is to write the script completly by yourself and possibly gain insights of my personal script. Please add your adjustments to the code as it may not use your coordinate system.

Introduction

The problem is an inversion problem. We have collected measures g. We have theoretical model K. The reason f is to be found such that

$$g \approx K(f). \tag{J1}$$

In tomography, the measures are transmitted rays through the object. These ray intensities are different from original intensity by reason f, that is, object's absorbtion and scattering properties [2].

The mathematic model of this work is attentuation law, which descibes ray to be attentuated depending it's path and attentuation distribution of the object to be tomographed.

Ray attentuation in a medium

Attentuatuon of a ray in a path δ follows attentuation law

$$\frac{dI}{ds} = -\mu I, \tag{1}$$

where $I = I(s)$ is rays intensitety in path's δ lenght s and $\mu = \mu(s)$ it the attentuation factor in that position.

The attentuation law (1) is separable differential equation, let us integrate to get:

$$\int_{I(0)}^{I(b)} \frac{1}{I} dI = \int_0^b -\mu ds, \tag{2}$$

where $b = \int_\delta ds$ is path's δ length. Let us numerically divide the path to elements, where $\mu(s)$ is approximately constant, we get

$$-\ln\frac{I(b)}{I(0)} = \sum_j \mu_j ds_j, \tag{3}$$

where j runs through all elements, μ_j is the attentuation factor in the element j ja ds_j rays travelled distance in element j.

We can build a vector f, an horizontal vector k ja a scalar g, such that

$$f = (\mu_1, \mu_2, ...), \tag{4}$$

$$k = (ds_1, ds_2, ...)^T, \tag{5}$$

$$g = -\ln\frac{I(b)}{I(0)}. \tag{6}$$

With number arrays (4)-(6), we can rewrite equation (3) to gain:

$$g = kf. \tag{7}$$

Calculating attentuation factors with Tikhonov regularization

If we measure many times following equation (7) we get a matrix equation

$$G = Kf, \tag{8}$$

where G is a vector, such that it's entry G_i follows eq. (6) so it is a measure $G_i = -\ln\frac{I(b)_i}{I(0)}$. K is a tomographical image matrix, consisting entries (5), both in current measure event. So $K_{ij} = (ds_j)_i$ where i is event number. In other words, K_{ij} is measure's i ray travel path j. Becouse of measure error or becouse of underdetermined measure amount, there may not find a unique vector f which agrees with (8). To estimate this vector, we use Tikhonov regularized least squares method. The estimate is

$$f \approx (K^TK + \alpha L_n^T L_n)^{-1} K^T G, \tag{9}$$

where L_n is n degree differenc ematrix of unit matrix (degree $n = 0$ makes L_n a unit matrix) and α some non-negative number [1]. If $\alpha = 0$ (9) is regular least squares approximation.

Condition on a Matrix

Matrix' K condition value $cond(K)$ is defined by following

$$cond(K) = \sigma_{max} : \sigma_{min}, \tag{K1}$$

where σ_{max} is matrix' K largest ja σ_{min} smallest singular value. Matrix' condition value. describes how vulnerable that as a inversion model is to errors in data. The more close the value is to unity, the better it behaves numerically [3].

Materials and methods

A device, shown in in Figure 2, that is, Am-241 γ -radiator (Americium isotope 241 high-frequency photon -radiator), is let to send rays to detector through an object, consisting a pattern of wooden cuboids, shown in figure (1). At the time of measures, the objects is covered with a plastic box, so it is impossible to tell the configuration by human eyesight. The detector is plugged in an amplifier, where it is redirected to a quantum counter. The detector and counter are manufactured by Ortec and counter by Canberra. Used electical tension ("voltage") were about 1 kV [one kilovolt]. The quantum counter calulated incoming γ -quantum amount in duration of 30 seconds. 88 measures are done, which of five partial series included "hand fan" measures (series 7-12). The another series of measures consisted 0, 90, 45, and -45 degree angle (series 1,2, 3+4 and 5+6), by figures 3 coordinate system.

The measure series' coordinates are tabulated it tables 1-12.

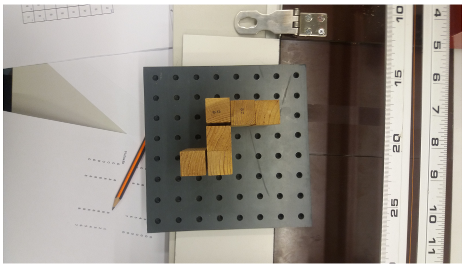


Figure 1: Opened object

Start-coord.	End-coord.
(0,5 ; 0)	(0,5 ; 8)
(1,5 ; 0)	(1,5 ; 8)
(2,5 ; 0)	(2,5 ; 8)
(3,5 ; 0)	(3,5 ; 8)
(4,5 ; 0)	(4,5 ; 8)
(5,5 ; 0)	(5,5 ; 8)
(6,5 ; 0)	(6,5 ; 8)
(7,5 ; 0)	(7,5 ; 8)

Table 1: serie 1

Start-coord.	End-coord.
(0 ; 0,5)	(8 ; 0,5)
(0 ; 1,5)	(8 ; 1,5)
(0 ; 2,5)	(8 ; 2,5)
(0 ; 3,5)	(8 ; 3,5)
(0 ; 4,5)	(8 ; 4,5)
(0 ; 5,5)	(8 ; 5,5)
(0 ; 6,5)	(8 ; 6,5)
(0 ; 7,5)	(8 ; 7,5)

Table 2: serie 2

Start-coord.	End-coord.
(0 ; 0)	(8 ; 8)
(0 ; 1)	(7 ; 8)
(0 ; 2)	(6 ; 8)
(0 ; 3)	(5 ; 8)
(0 ; 4)	(4 ; 8)
(0 ; 5)	(3 ; 8)
(0 ; 6)	(2 ; 8)
(0 ; 7)	(1 ; 8)

Table 3: serie 3

Start-coord.	End-coord.
(7 ; 0)	(8 ; 1)
(6 ; 0)	(8 ; 2)
(5 ; 0)	(8 ; 3)
(4 ; 0)	(8 ; 4)
(3 ; 0)	(8 ; 5)
(2 ; 0)	(8 ; 6)
(1 ; 0)	(8 ; 7)

Table 4: serie 4

Start-coord.	End-coord.
(0 ; 1)	(1 ; 0)
(0 ; 2)	(2 ; 0)
(0 ; 3)	(3 ; 0)
(0 ; 4)	(4 ; 0)
(0 ; 5)	(5 ; 0)
(0 ; 6)	(6 ; 0)
(0 ; 7)	(7 ; 0)
(0 ; 8)	(8 ; 0)

Table 5: serie 5

Start-coord.	End-coord.
(1 ; 8)	(8 ; 1)
(2 ; 8)	(8 ; 2)
(3 ; 8)	(8 ; 3)
(4 ; 8)	(8 ; 4)
(5 ; 8)	(8 ; 5)
(6 ; 8)	(8 ; 6)
(7 ; 8)	(8 ; 7)

Table 6: serie 6

Start-coord.	End-coord.
(0 ; 0)	(1 ; 8)
(0 ; 0)	(2 ; 8)
(0 ; 0)	(3 ; 8)
(0 ; 0)	(4 ; 8)
(0 ; 0)	(5 ; 8)
(0 ; 0)	(6 ; 8)
(0 ; 0)	(7 ; 8)

Table 7: serie 7

Start-coord.	End-coord.
(0 ; 0)	(8 ; 7)
(0 ; 0)	(8 ; 6)
(0 ; 0)	(8 ; 5)
(0 ; 0)	(8 ; 4)
(0 ; 0)	(8 ; 3)
(0 ; 0)	(8 ; 2)
(0 ; 0)	(8 ; 1)

Table 8: serie 8

Start-coord.	End-coord.
(8 ; 0)	(0 ; 1)
(8 ; 0)	(0 ; 2)
(8 ; 0)	(0 ; 3)
(8 ; 0)	(0 ; 4)
(8 ; 0)	(0 ; 5)
(8 ; 0)	(0 ; 6)
(8 ; 0)	(0 ; 7)

Table 9: serie 9

Start-coord.	End-coord.
(8 ; 0)	(1 ; 8)
(8 ; 0)	(2 ; 8)
(8 ; 0)	(3 ; 8)
(8 ; 0)	(4 ; 8)
(8 ; 0)	(5 ; 8)
(8 ; 0)	(6 ; 8)
(8 ; 0)	(7 ; 8)

Table 10: serie 10

Start-coord.	End-coord.
(0 ; 8)	(1 ; 0)
(0 ; 8)	(2 ; 0)
(0 ; 8)	(3 ; 0)
(0 ; 8)	(4 ; 0)
(0 ; 8)	(5 ; 0)
(0 ; 8)	(6 ; 0)
(0 ; 8)	(7 ; 0)

Table 11: serie 11

Start-coord.	End-coord.
(0 ; 8)	(8 ; 1)
(0 ; 8)	(8 ; 2)
(0 ; 8)	(8 ; 3)
(0 ; 8)	(8 ; 4)
(0 ; 8)	(8 ; 5)
(0 ; 8)	(8 ; 6)
(0 ; 8)	(8 ; 7)

Table 12: serie 12

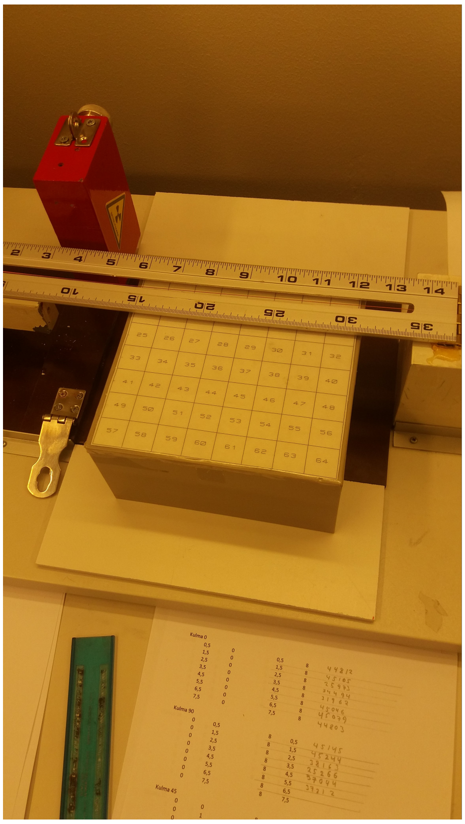


Figure 2: Measure device

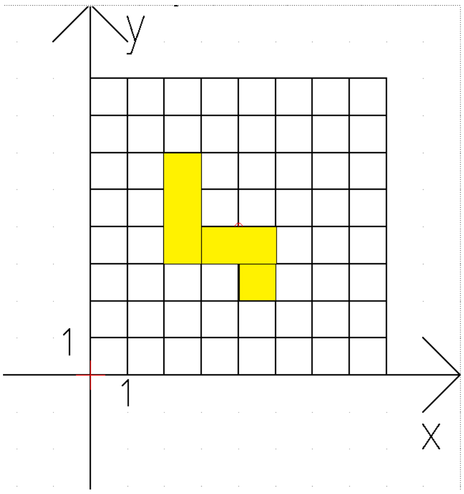
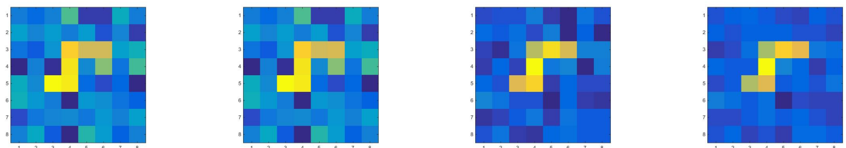


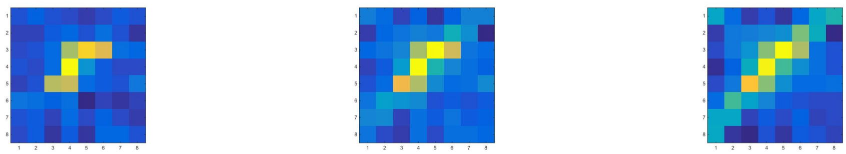
Figure 3: Measure orientation

Data-analysis and Results

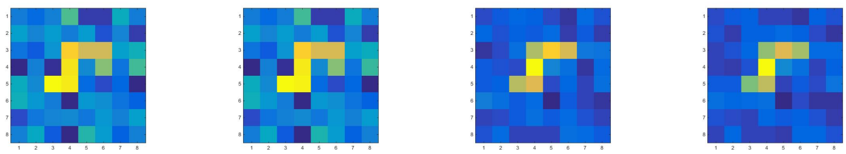
The object is divided to 8x8=64 square elements. Full imagining matrix $K \in \mathbb{R}^{88\times 64}$ is built. By eq. (9), the attentuation distribution is calculated, with different values of α and n. We see the calculated images in figureserie 4. The condition number was about 240. By my eyesight, the figure looks clearest with values $\alpha = 10^{-1}$ ja $n = 2$. New images with these values are build, this time we reduce the measure vectors. First of these images included only series 1-6 (cond: $9.5 \cdot 10^{15}$), another 7-12 (cond: $1.5 \cdot 10^{16}$) ja third 1-2 & 11-12 (cond: $1.8 \cdot 10^{16}$). Fourth image had almost all series, 1-11 (cond: 4321). We see these images in figureserie 5.



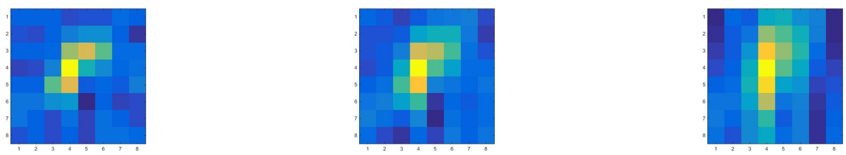
(a) $\alpha = 10^{-\infty}$, $n = 0$ (b) $\alpha = 10^{-5}$, $n = 0$ (c) $\alpha = 10^{-2}$, $n = 0$ (d) $\alpha = 10^{-1}$, $n = 0$



(e) $\alpha = 10^{0}$, $n = 0$ (f) $\alpha = 10^{1}$, $n = 0$ (g) $\alpha = 10^{2}$, $n = 0$

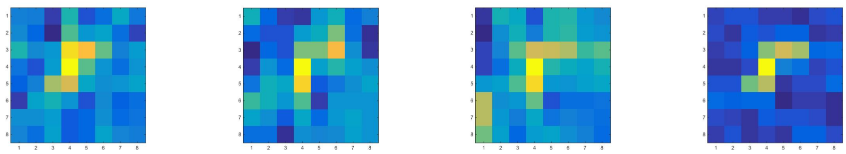


(h) $\alpha = 10^{-\infty}$, $n = 2$ (i) $\alpha = 10^{-5}$, $n = 2$ (j) $\alpha = 10^{-2}$, $n = 2$ (k) $\alpha = 10^{-1}$, $n = 2$



(l) $\alpha = 10^{0}$, $n = 2$ (m) $\alpha = 10^{1}$, $n = 2$ (n) $\alpha = 10^{2}$, $n = 2$

Figure 4: Eq. (9) reconstructed images with full $\mathbb{R}^{88x64}$ imaging matrix, with different α and n values.



(a) Series 1-6 (b) Series 7-12 (c) Series 1, 2, 11 and 12 (d) Series 1-11

Figure 5: Eq. (9) reconstructed images with different imaging matrices, $\alpha = 10^{-1}$ and $n = 2$.

About used MATLAB code

Two **.mat** fires are read. The matrix K is initialised. In nested loops the entries are calculated using standard Pythagorean theorem. There are nCr(4,2) possibilities, i.e. four sides and ray enters one of them, and leaves another. So six **if** conditions are written. In version 2 this is done by a loop. In original version, they are explicitly written. Tikhonov's method is applied and figure shown.

Discussion

We see that the reconstuction show the wooden cuboids, so the method works in this situation. Artifacts from measure error are low with Tikhonov's values $\alpha \in [10^{-2}, 10^{-1}]$. If α value is raised, the image blurs. But if it near zero, artifacts can be seen. The whole series' matrix' condition is relatively small, compared to under determined ones. Also the image built with almost complete matrix, fig. (5 d)), had a relatively small condition number. Other figures had very large condition values becouse of under determicy. We can conclude from this that condition number is a useful tool to see the numerical behauvior and determicy. Tikhonov's regularization method was important tool in reconstructing the figures.

References

[1] Reichel, L., Ye, Q.: Simple square smoothing regularization operators. Electron. Trans. Numer. Anal. 33, 63-83 (2009)
[2] Käänteisongelmat, Jari Kaipio (2002)
[3] MathWorks®Documentation: cond, referenced in 17.11.2016, URL: `https://se.mathworks.com/help/matlab/ref/cond.html?requestedDomain=www.mathworks.com`

Appendix: MATLAB code

```
% Mika L
%reconstructs picture from tomografical experiement
load('in_out_coordinate_data.mat')
load('radioactivity_count_data.mat')

%grid size
gsize=[8 8]';

%put coordinate data into a matrix
MitM=[
Mit1;%0 directly
Mit2;%on straight 90 angle
Mit3;
Mit4;%on angle 45
Mit5;
Mit6;%ending,-45
Mit7;
Mit8;%"hand fan"1
Mit9;
Mit10;%"hand fan"2
Mit11;
Mit12;%"hand fan"3


];


rdivec=[
rdi1;
rdi2;
rdi3;
rdi4;
rdi5;
rdi6;
rdi7;
rdi8;
rdi9;
rdi10;
rdi11;
rdi12;


];

%solution is based to taking the logarithm:
logvec=-log(rdivec/max(rdivec));

msize=size(MitM);

%init pic matrix
KuvM=zeros(msize(1),gsize(1)*gsize(2));



% 3 nested loops, inside nCr(4,2)=6 condition-checks
%with 8 conditions
```

```
for k=1:msize(1)

A=[MitM(k,1) MitM(k,2)]';
B=[MitM(k,3) MitM(k,4)]';


M=zeros(gsize(1),gsize(2));

%loop every pixel
for m=1:gsize(1)
for n=1:gsize(2)

pa=A+((m-A(1))/(B(1)-A(1)))*(B-A);%starting point grid cell row
pb=A+((m-1-A(1))/(B(1)-A(1)))*(B-A);%starting point grid cell col
pc=A+((n-A(2))/(B(2)-A(2)))*(B-A);%ending point grid cell row
pd=A+((n-1-A(2))/(B(2)-A(2)))*(B-A);%ending point grid cell col


%check every nCr possibility, if radiation hits a grid, calculate
%geometrical path length in that grid cell
%4 sides and in+out so nCr=6
if m-1<=pa(1) && pa(1)<=m && n-1<=pa(2) && pa(2)<=n && m-1<=pb(1) && pb(1)<=m && n-1<=pb(2) && pb
    if sqrt((pa(1)-pb(1))^2+(pa(2)-pb(2))^2)>0
    M(m,n)=sqrt((pa(1)-pb(1))^2+(pa(2)-pb(2))^2);
    end
end

if m-1<=pa(1) && pa(1)<=m && n-1<=pa(2) && pa(2)<=n && m-1<=pc(1) && pc(1)<=m && n-1<=pc(2) && pc
    if sqrt((pa(1)-pc(1))^2+(pa(2)-pc(2))^2)>0
    M(m,n)=sqrt((pa(1)-pc(1))^2+(pa(2)-pc(2))^2);
    end
end


if m-1<=pa(1) && pa(1)<=m && n-1<=pa(2) && pa(2)<=n && m-1<=pd(1) && pd(1)<=m && n-1<=pd(2) && pd
    if sqrt((pa(1)-pd(1))^2+(pa(2)-pd(2))^2)>0
    M(m,n)=sqrt((pa(1)-pd(1))^2+(pa(2)-pd(2))^2);
    end
end

if m-1<=pb(1) && pb(1)<=m && n-1<=pb(2) && pb(2)<=n && m-1<=pc(1) && pc(1)<=m && n-1<=pc(2) && pc
    if sqrt((pb(1)-pc(1))^2+(pb(2)-pc(2))^2)>0
    M(m,n)=sqrt((pb(1)-pc(1))^2+(pb(2)-pc(2))^2);
    end
end


if m-1<=pb(1) && pb(1)<=m && n-1<=pb(2) && pb(2)<=n && m-1<=pd(1) && pd(1)<=m && n-1<=pd(2) && pd

    if sqrt((pb(1)-pd(1))^2+(pb(2)-pd(2))^2) >0
    M(m,n)=sqrt((pb(1)-pd(1))^2+(pb(2)-pd(2))^2);
    end

end

if m-1<=pc(1) && pc(1)<=m && n-1<=pc(2) && pc(2)<=n && m-1<=pd(1) && pd(1)<=m && n-1<=pd(2) && pd

    if sqrt((pc(1)-pd(1))^2+(pc(2)-pd(2))^2)>0
    M(m,n)=sqrt((pc(1)-pd(1))^2+(pc(2)-pd(2))^2);
    end

end
```

```
end
end

Mres=reshape(M,[gsize(1)*gsize(2),1]);
KuvM(k,:)=Mres;

end



idim=size(KuvM'*KuvM);
I=eye(idim(1));


%%%%Andrei Tikhonov's regularization
L0=I;
L1=diff(I);
L2=diff(diff(I));

tikho0=L0'*L0;
tikho1=L1'*L1;
tikho2=L2'*L2;


alpha=10^(-1);
tikho=tikho2;


picvec=(KuvM'*KuvM+alpha*tikho)\KuvM'*logvec;
%matlab operand x\y means inverse(x)*y
pic=reshape(picvec,gsize(1),gsize(2));
imagesc(pic);
title('Reconstructed tomographical image')
axis image
```