Tomographic X-ray data of a cross phantom

S. Latva-Äijö*, A. Meaney†, and S. Siltanen‡

26. lokakuuta 2018

Tiivistelmä

This is the documentation of the tomographic X-ray data of a dynamic cross phantom made available here. The data can be freely used for scientific purposes with appropriate references to the data and to this document in http://arxiv.org/. The data set consists of (1) the X-ray sinogram with 16 or 30 time frames (depending on the resolution) of 2D slices of the phantom and (2) the corresponding static and dynamic measurement matrices modeling the linear operation of the X-ray transform. Each of these sinograms was obtained from a measured 360-projection fan-beam sinogram by down-sampling and taking logarithms.

This new version of the data set contains added files `DataStatic_560x60.mat` (with 80 time frames) and `DataStatic_1120x60.mat` (with 230 time frames). You can run these files with the code `example_2`, which computes a Tikhonov regularized reconstruction using conjugate gradient algorithm.

1 Introduction

The main idea behind the project was to create real CT measurement data for testing sparse data and dynamic data tomography algorithms. A simple target which is changing in time was prepared by crossing aluminum stick and graphite stick so that sticks are in slightly different angles compared to each other. The system was stabilized using candle

*Department of Mathematics and Statistics, University of Helsinki, Finland (salla.latva-aijo@helsinki.fi)

†Department of Mathematics and Statistics, University of Helsinki, Finland (alexander.meaney@helsinki.fi)

‡Department of Mathematics and Statistics, University of Helsinki, Finland (samuli.siltanen@helsinki.fi)

wax. See Figure 2 of the preparation phase of the phantom. When the target rotates and we observe it along the z-axis, the cross sections of the sticks move toward each other or appart from each other in every time step. The time-dependent target is challenge for sparse dynamic tomography applications. The CT data in this data set has been used for example with the modified level set reconstruction method, see FIPS Computational Blog.

2 Contents of the data set

The data set contains the following MATLAB[1] data files:

- `DataStatic_128x60.mat`,
- `DataStatic_128x15.mat`,
- `DataDynamic_128x60.mat`,
- `DataDynamic_128x15.mat`,
- `DataStatic_256x60.mat`,
- `DataStatic_256x15.mat`,
- `DataDynamic_256x60.mat`,
- `DataDynamic_256x15.mat`,
- `DataStatic_560x60.mat`,
- `DataStatic_1120x60.mat`,
- `FullSizeSinogram.mat` and
- `GroundTruthReconstruction.mat`.

The first four of these files contain CT sinogram and the corresponding measurement matrix with the resolution 128×128 as spatial resolution and 16 as a temporal resolution in 3D. In the last four files the contents are the same, but spatial resolution is 256×256 and temporal resolution in 3D is 30 time frames. The word `Static` in the file name means that the data leads to reconstructions with the same geometry in every time instance. The word `Dynamic` in the file name means that the projection angles shift by one degree in every time step, for example, if the angles for time t_1 are $[1 \quad 13 \quad 25 \cdots]$, then for time t_2, the angles are $[2 \quad 14 \quad 26 \cdots]$. The detailed contents of every file are listed below.

`DataStatic_128x60.mat` contains the following variables:

[1]MATLAB is a registered trademark of The MathWorks, Inc.

1. Sparse matrix `A` of size 134 400 × 262 144 sparse double; measurement matrix.
2. Matrix `sinogram` of size 140 × 960; sinogram (60 projections out of full 360 degree circle).

`DataStatic_128x15.mat` contains the following variables:

1. Sparse matrix `A` of size 33 600 × 262 144; measurement matrix.
2. Matrix `sinogram` of size 140 × 240; sinogram (15 projections).

`DataDynamic_128x60.mat` contains the following variables:

1. Sparse matrix `A` of size 134 400 × 262 144; measurement matrix.
2. Matrix `m` of size 140 × 960; sinogram (60 projections).

`DataDynamic_128x15.mat` contains the following variables:

1. Sparse matrix `A` of size 33 600 × 252 144; measurement matrix.
2. Matrix `sinogram` of size 140 × 240; sinogram (15 projections).

`DataStatic_256x60.mat` contains the following variables:

1. Sparse matrix `A` of size 504 000 × 1 966 080; measurement matrix.
2. Matrix `sinogram` of size 280 × 1800; sinogram (60 projections).

`DataStatic_256x15.mat` contains the following variables:

1. Sparse matrix `A` of size 126 000 × 1 966 080; measurement matrix.
2. Matrix `sinogram` of size 280 × 450; sinogram (15 projections).

`DataDynamic_256x60.mat` contains the following variables:

1. Sparse matrix `A` of size 504 000 × 1 966 080; measurement matrix.
2. Matrix `sinogram` of size 280 × 1800; sinogram (60 projections).

`DataDynamic_256x15.mat` contains the following variables:

1. Sparse matrix `A` of size 126 000 × 1 966 080; measurement matrix.
2. Matrix `sinogram` of size 280 × 450; sinogram (15 projections).

`DataStatic_560x60.mat` contains the following variables:

1. Sparse matrix `A` of size 33 600 × 176 400; measurement matrix.
2. Matrix `sinogram` of size 560 × 4 800; sinogram (60 projections).

`DataStatic_1120x60.mat` contains the following variables:

1. Sparse matrix `A` of size $67\,200 \times 1\,048\,576$; measurement matrix.
2. Matrix `sinogram` of size 1120×13800; sinogram (60 projections).

`FullSizeSinogram.mat` contains the following variables:

1. Matrix `sinogram` of size 2240×360; original (measured) sinogram of 360 projections.

`GroundTruthReconstruction.mat` contains the following variables:

1. Matrix `GroundTruthReconstruction.mat` of size 2240×360; a high-resolution filtered backprojection reconstruction computed from the larger sinogram of 360 projections of the cross phantom (”ground truth”). See Figure 6.

Details on the X-ray measurements are described in Section 3 below.

The model for the CT problem is

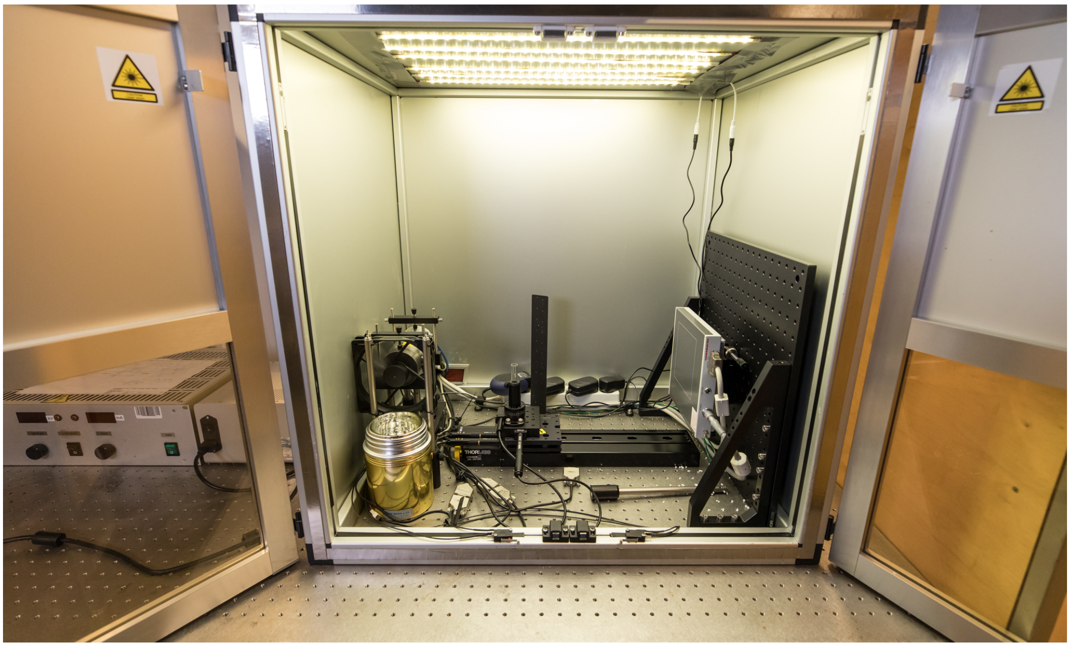
$$\mathtt{A} * \mathtt{x} = \mathtt{m}(:), \tag{1}$$

where `m(:)` denotes the standard vector form of matrix `m` in MATLAB (`m` corresponds to matrix `sinogram`) and `x` is the reconstruction in vector form. In other words, the reconstruction task is to find a vector `x` that (approximately) satisfies (1) and possibly also meets some additional regularization requirements. A demonstration of the use of the data is presented in Section 4 below.

3 X-ray measurements

The data in the sinogram is X-ray tomographic (CT) data of a 2D cross-section of the dynamic phantom built from sticks and candle wax and measured with a custom built CT device shown in Figure 1.

- The X-ray tube is a model XTF5011 manufactured by Oxford Instruments. This model is no longer sold by Oxford Instruments, although they have newer, similar models available. The tube uses a molybdenum ($Z = 42$) target.
- The rotation stage is a Thorlabs model CR1/M-27.
- The flat panel detector is a Hamamatsu Photonics C7942CA-22. The active area of the at panel detector is 120 mm $\times$120 mm. It consists of a 2400×2400 array of 50 μm pixels. According to the manufacturer, the number of active pixels is 2240×2344. However, the image files actually generated by the camera were 2240×2368 pixels in size.

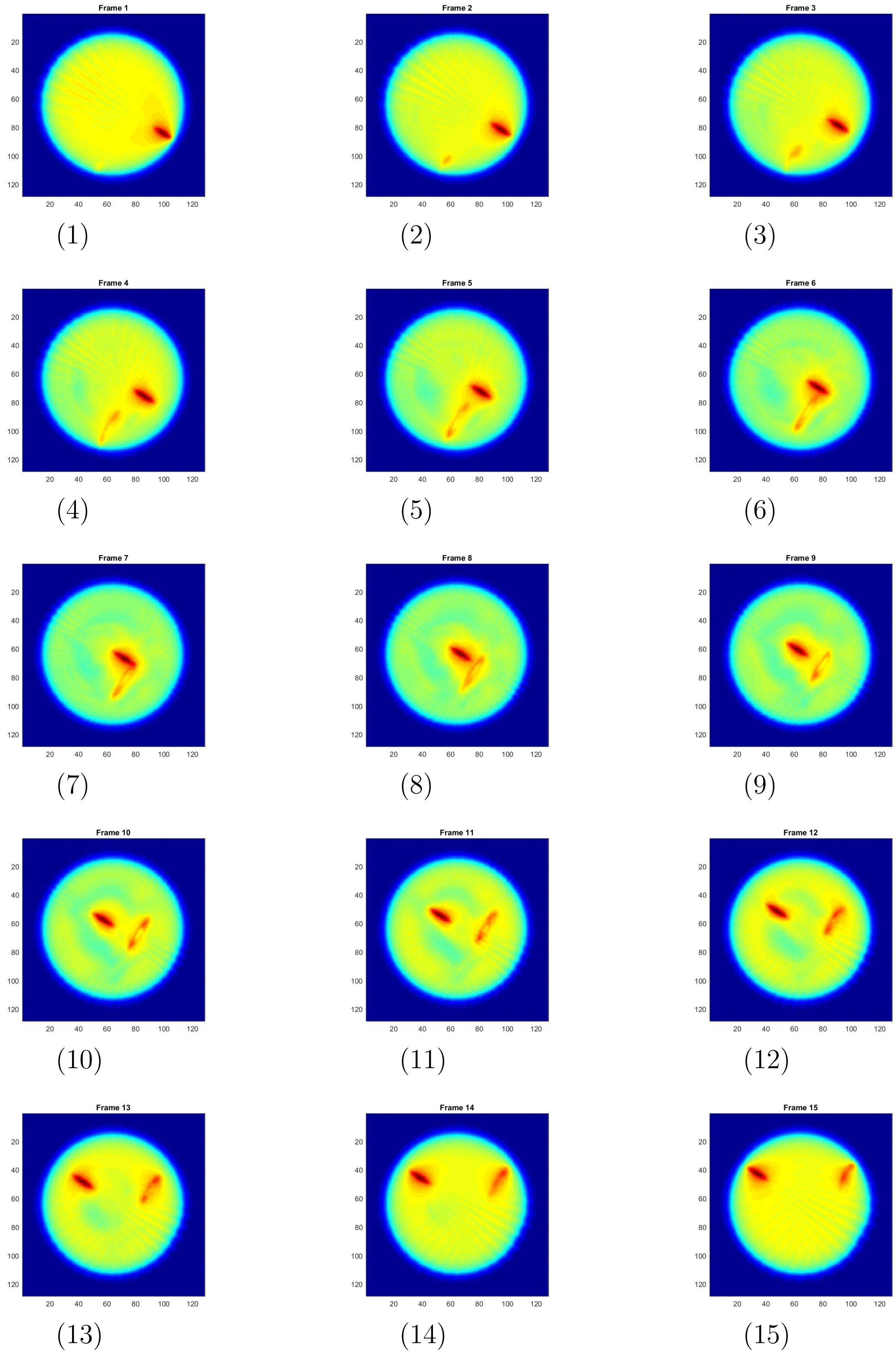


Kuva 1: The custom-made measurement device at University of Helsinki. Figure was taken by Markus Juvonen.

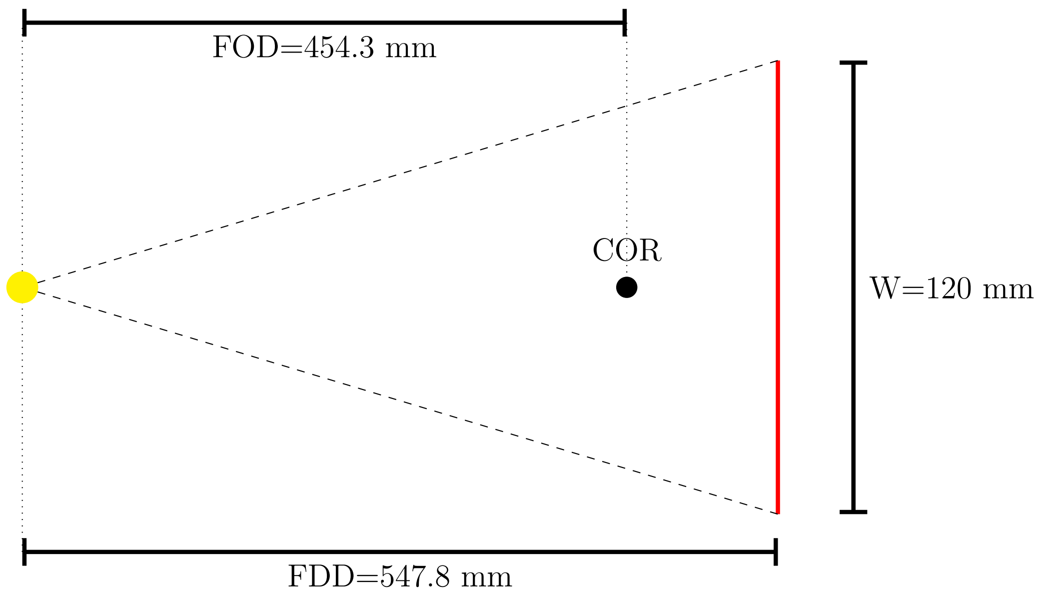
The measurement setup was assembled in 2015 by Alexander Meaney as an MSc thesis project [1]. The setup is illustrated in Figure 1 and the measurement geometry is shown in Figure 4. A set of 360 cone-beam projections with resolution 2240×2368 and the angular step of one (1) degree was measured. The exposure time was 1000 ms (i.e., one second). The X-ray tube acceleration voltage was 50 kV and tube current 0.9 mA. See Figure 2 for example of the resulting projection image.

From the 2D projection images, the 230 rows from the projection image were selected to form sinograms of the dynamic phantom of resolution 2240×360. These sinograms were further down-sampled by binning, taken logarithms and normalizing to obtain the `sinogram` in all the files specified in Section 2. The organization of the pixels in the sinograms and the reconstructions is illustrated in Figure 5.

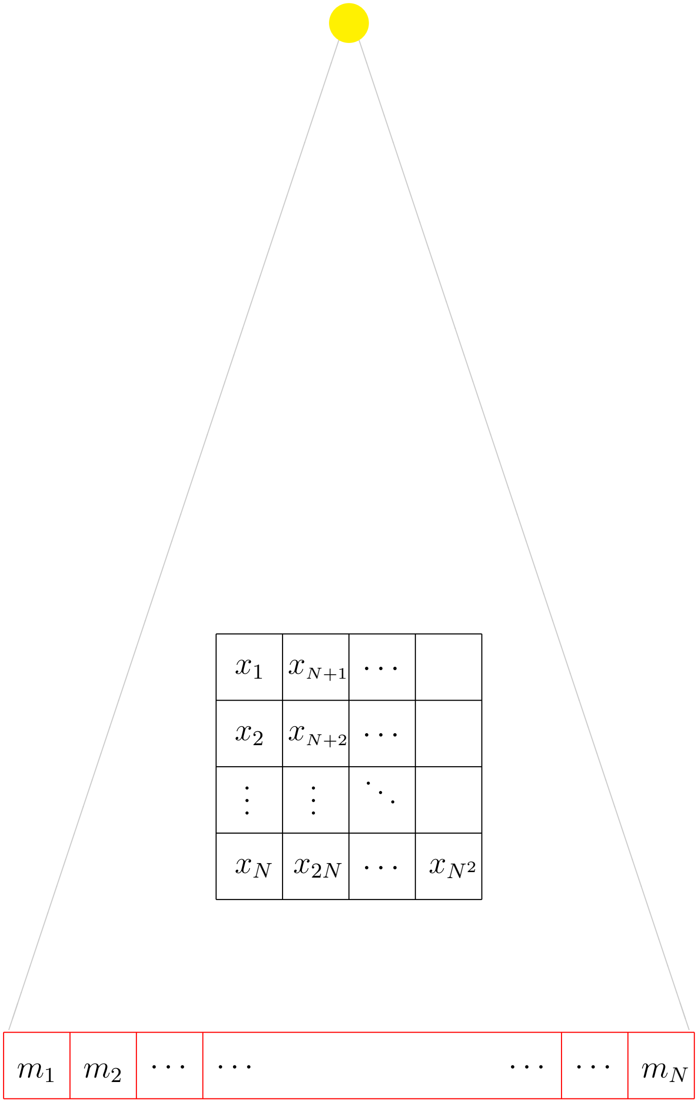
Kuva 2: *Left*: The phantom in the preparation phase, when the positions of the sticks were set. The darker stick is a graphite taken from a pen. The aluminum stick was made by rolling aluminum sheet, which explains its irregular shape. It is also hollow and contains air, which can been seen in the resulting reconstructions. *Middle*: The ready-made phantom on the sample holder, which can be attached to a computer-controlled rotator platform on the CT device, see Figure 1. *Right*: Two examples of the resulting 2D projection images.



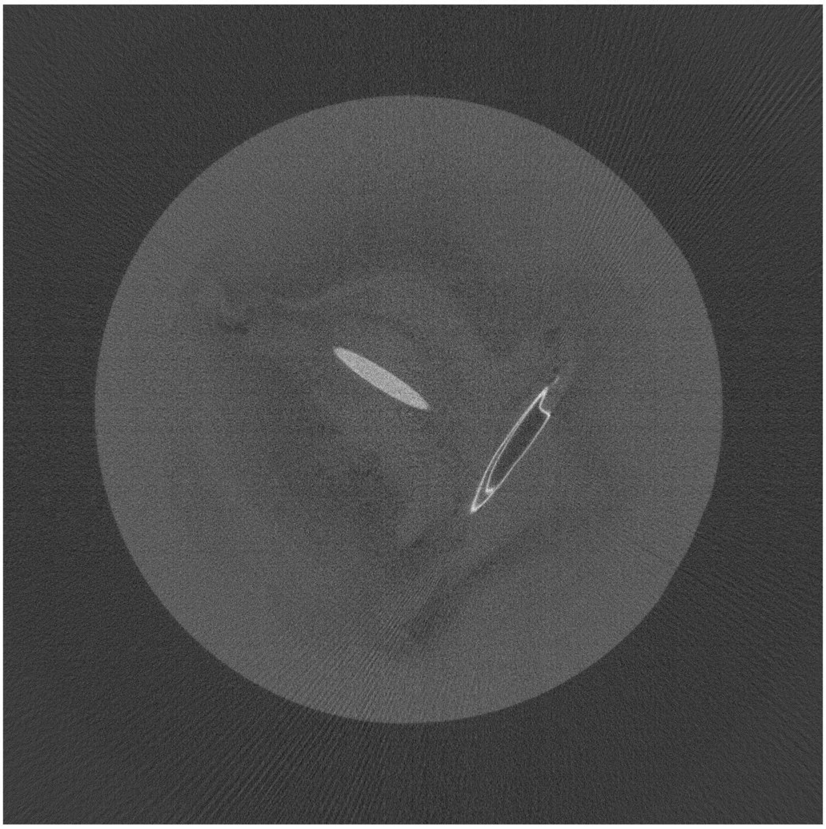
Kuva 3: Tikhonov reconstructions of 15 different example frames showing the movement of the cross sections of the aluminum and graphite sticks.



Kuva 4: Geometry of the measurement setup. Here FOD and FDD denote the focus-to-object distance and the focus-to-detector distance, respectively; the black dot COR is the center-of-rotation. The width of the detector (*i.e.*, the red thick line) is denoted by W. The yellow dot is the X-ray source. To increase clarity, the x-axis and y-axis in this image are not in scale.



Kuva 5: The organization of the pixels in the sinograms $\mathtt{m} = [m_1, m_2, \ldots, m_{60K}]^T$ and reconstructions $\mathtt{x} = [x_1, x_2, \ldots, x_{N^2}]^T$ with $N = 256$. The picture shows the organization for the first projection; after that in the full angular view case, the target takes 6 degree steps counter-clockwise (or equivalently the source and detector take 6 degree steps clockwise) and the following columns of $\mathtt{m}$ are determined in an analogous manner.



Kuva 6: The high-resolution filtered back-projection reconstruction of the time-dependent cross phantom computed from 360 projections.

4 Examples of using the data

The following MATLAB codes demonstrate how to use the data. The codes are also provided as the separate MATLAB script files `example_1.m` and `example_2.m`. First example assumes the data file used is included in the same directory with the script file. To run second example (for 80/120 time frame data) you also need to have function blkmulti in the same directory with the script file.

example_1.m

```
%% Tikhonov regularized reconstruction, example 1
% Compute a Tikhonov regularized reconstruction using
% conjugate gradient algorithm pcg.m

% Load the measurement matrix and the sinogram from some of the files
load DataStatic_256x60 A sinogram
%load DataDynamic_128x15 A sinogram

m = sinogram;
% Compute a Tikhonov regularized reconstruction using
% conjugate gradient algorithm pcg.m

% Define the number of slices
T = 30; %(16 or 30 depending on resolution)
N     = sqrt(size(A,2)/T);
alpha = 10; % regularization parameter
fun   = @(x) A.'*(A*x)+alpha*x;
b     = A.'*m(:);
x     = pcg(fun,b);

% Compute a Tikhonov regularized reconstruction from only
% 10 projections
[mm,nn] = size(m);
ind     = [];
for iii=1:nn/6
    ind = [ind,(1:mm)+(6*iii-6)*mm];
end
m2    = m(:,1:6:end);
A     = A(ind,:);
alpha = 10; % regularization parameter
fun   = @(x) A.'*(A*x)+alpha*x;
b     = A.'*m2(:);
```

```
x2    = pcg(fun,b);

% Take a look at the sinograms of the first slice and the reconstructions
figure
subplot(2,2,1)
imagesc(m(:,1:size(m,2)/T))
colormap gray
axis square
axis off
title('Sinogram, 60 projections')
subplot(2,2,3)
imagesc(m2(:,1:size(m2,2)/T))
colormap gray
axis square
axis off
title('Sinogram, 10 projections')
% Reshape the reconstruction
x = reshape(x,N,N,T);
x2 = reshape(x2,N,N,T);
% Show the first slice reconstruction
subplot(2,2,2)
imagesc(imrotate(x(:,:,1),-98,'bilinear','crop'))
colormap gray
axis square
axis off
title({'Tikhonov reconstruction,'; '60 projections'})
subplot(2,2,4)
imagesc(imrotate(x2(:,:,1),-98,'bilinear','crop'))
colormap gray
axis square
axis off
title({'Tikhonov reconstruction,'; '10 projections'})
```

example_2.m

```
%% Tikhonov regularized reconstruction, example 2
% Compute a Tikhonov regularized reconstruction using
% conjugate gradient algorithm pcg.m
% To run this code, you need your data file and the auxiliary function
% blkmulti.m to be in the same folder.
```

```
% Load the data
% load DataStatic_1120x60 A sinogram or load DataStatic_560x60 A sinogram
load DataStatic_560x60 A sinogram
m = sinogram;
m = m(:);
tt = 80; % number of time steps (80 or 230)
%%
% Initial guess for u
u = zeros(size(A,2)*tt,1);
ss = sqrt(size(u,1)/tt);
u = reshape(u,[ss,ss,tt]);

alpha = 0.5; % regularization parameter

ht = 1; % adjusts the amount of regularization in temporal direction

% First iteration
graddu2 = [];
graddu2=(8+4/ht)*u-2*([u(2:end,:,:);zeros(1,ss,tt)]+[u(:,2:end,:),...
zeros(ss,1,tt)]+[zeros(1,ss,tt);u(1:end-1,:,:)]+[zeros(ss,1,tt),...
u(:,1:end-1,:)])-(2/ht)*(cat(3,u(:,:,2:end),zeros(ss,ss,1))+...
cat(3,zeros(ss,ss,1),u(:,:,1:end-1)));
disp('Creating gradient function...');
graddu2 = graddu2(:);
u = u(:);

gradF1 = 2*blkmulti(A',blkmulti(A,u))-2*blkmulti(A',m)+alpha*graddu2;
disp('Gradient function F1 ready!');
%%
lam = .0001; % first step size

oldu = u;
u = max(0,u-lam*gradF1);

% Iterate
disp('Iterating... ' );
k = 3; % number of iterations
for l = 1:k
graddu2 = [];
u = reshape(u,[ss,ss,tt]);
graddu2=(8+4/ht)*u-2*([u(2:end,:,:);zeros(1,ss,tt)]+[u(:,2:end,:),...
zeros(ss,1,tt)]+[zeros(1,ss,tt);u(1:end-1,:,:)]+[zeros(ss,1,tt),...
u(:,1:end-1,:)])-(2/ht)*(cat(3,u(:,:,2:end),zeros(ss,ss,1))+...
```

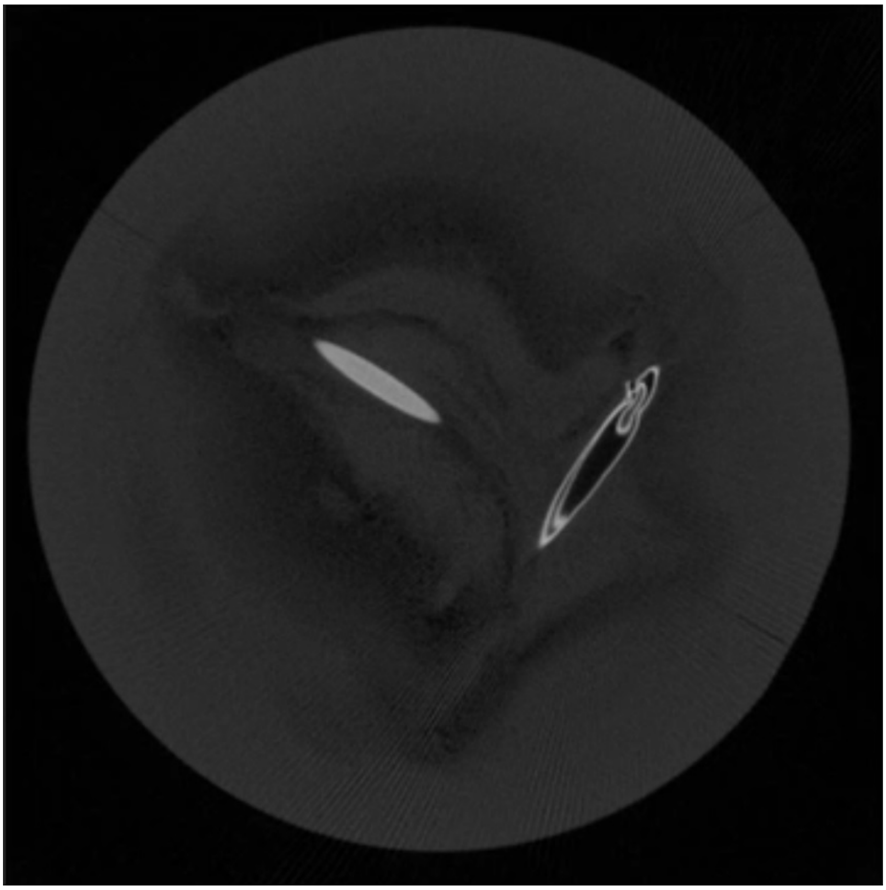
```
cat(3,zeros(ss,ss,1),u(:,:,1:end-1)));
graddu2 = graddu2(:);

u = u(:);
oldgradF1 = gradF1;
gradF1 = 2*blkmulti(A',blkmulti(A,u))-2*blkmulti(A',m)+alpha*graddu2;
lam = (u-oldu)'*(u-oldu)/((u-oldu)'*(gradF1-oldgradF1));
oldu = u;
u = max(0,u-lam*gradF1);
disp([num2str(l),'/' num2str(k)]);
end
disp('Iteration ready!');

%% Show reconstruction
u = reshape(u,[ss,ss,tt]);
for ii = 1:tt
imagesc(u(:,:,ii));
title(['Frame ',num2str(ii)]);
axis square;
colormap jet;
%colorbar
pause(0.2);
end
%% Normalize and watch as a black and white video
% u = u - min(u(:));
% u = u ./ max(u(:));
% implay(u);
```

blkmulti.m

```
% Auxiliary function for matrix multiplication
function res = blkmulti(A0,b)
res = [];
n = size(A0,2);
for ii = 1:(size(b,1)/n)
    res = [res;A0*b(((ii-1)*n+1):ii*n)];
end
```

5 3D reconstruction

The video below is three-dimensional ”ground truth”reconstruction of the time-dependent cross phantom target. This reconstruction is made from the original measured data (360 angles) and reconstructed using the FDK-cuda algorithm of the ASTRA toolbox [2] [2] [3].



[2]The ASTRA Toolbox is open source under the GPLv3 license.

Viitteet

[1] Alexander Meaney et al., *Design and construction of an X-ray computed tomography imaging system*, (2015).

[2] W. van Aarle, W. J. Palenstijn, J. Cant, E. Janssens, F. Bleichrodt, A. Dabravolski, J. De Beenhouwer, K. J. Batenburg, and J. Sijbers, *Fast and Flexible X-ray Tomography Using the ASTRA Toolbox*, Optics Express, 24(22), 25129-25147, (2016), http://dx.doi.org/10.1364/OE.24.025129

[3] W. van Aarle, W. J. Palenstijn, J. De Beenhouwer, T. Altantzis, S. Bals, K. J. Batenburg, and J. Sijbers, *The ASTRA Toolbox: A platform for advanced algorithm development in electron tomography*, Ultramicroscopy, 157, 35-47, (2015), http://dx.doi.org/10.1016/j.ultramic.2015.05.002