Ring artifact removal method using GPU-based FDK reconstruction for cone beam CT

Zsolt Adam Balogh

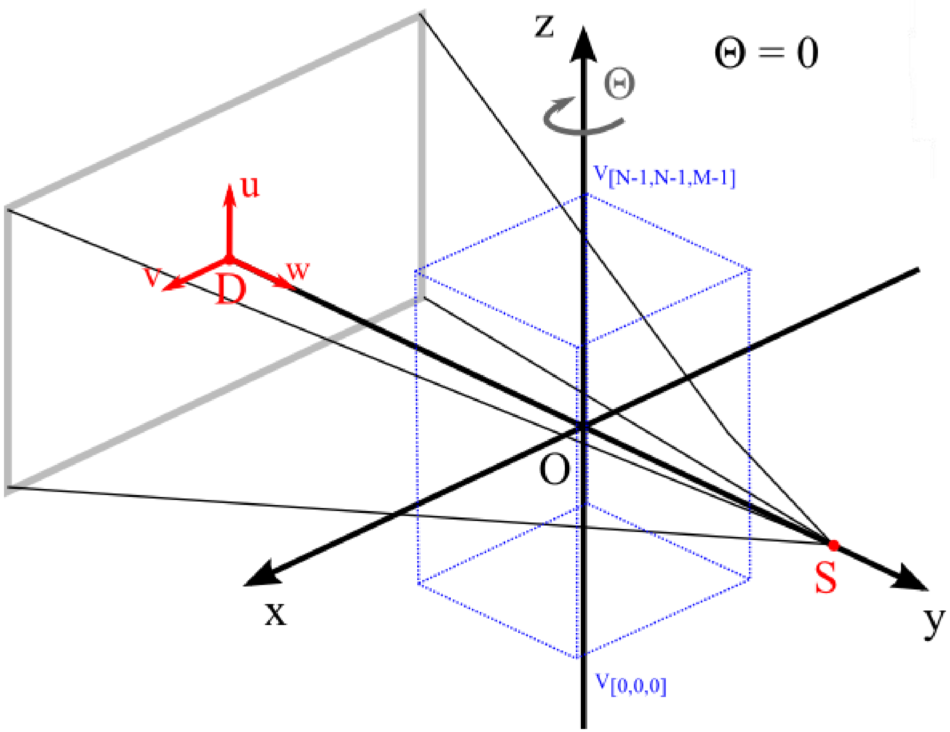
Zsolt.Balogh@mediso.com

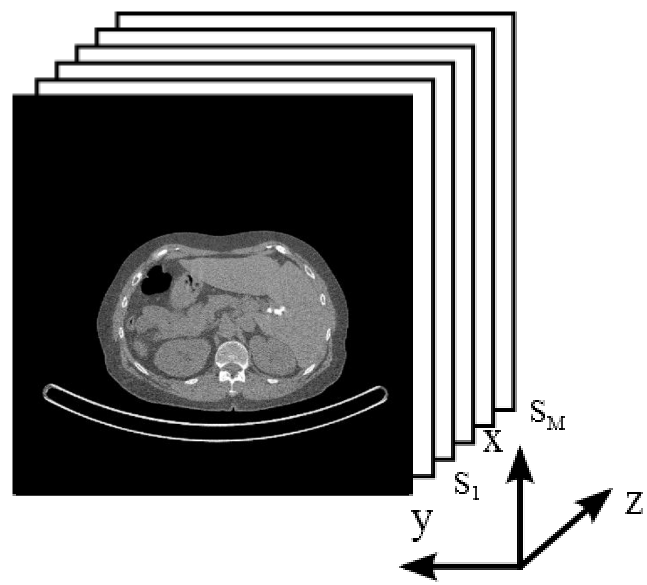
R & D Mediso Ltd.

2-3 June 2016 Budapest, Hungary

GPU DAY 2016 - THE FUTURE OF MANY-CORE COMPUTING IN SCIENCE

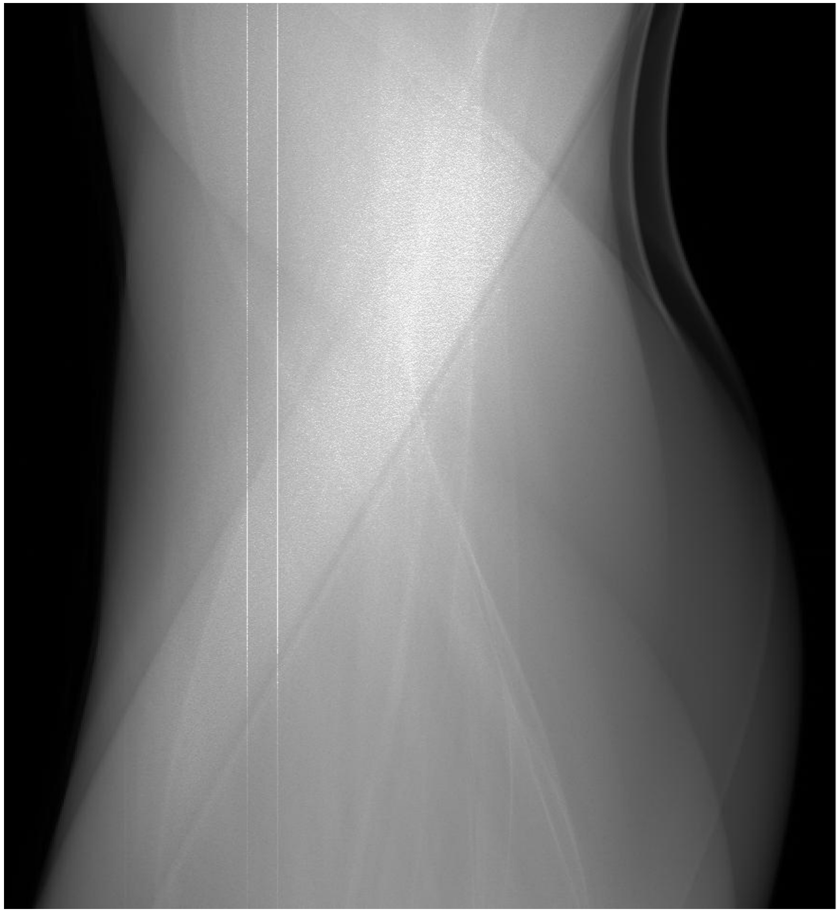
Computed Tomography





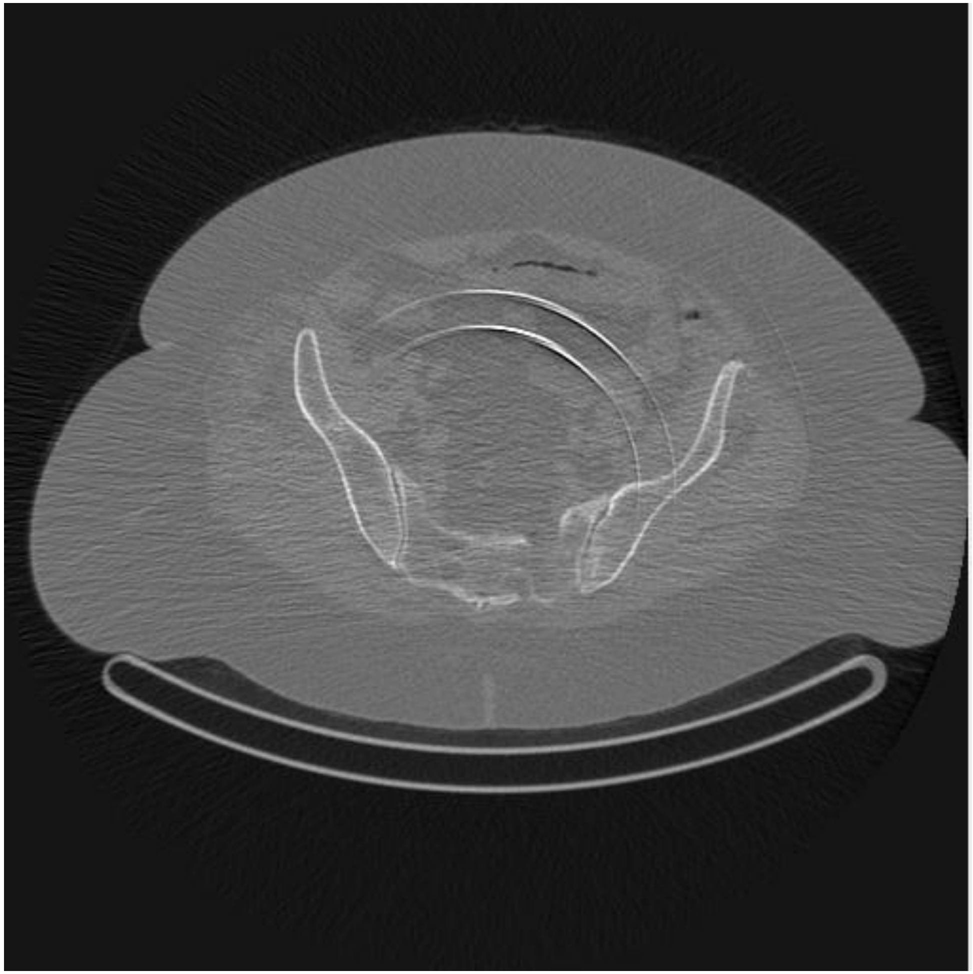
Ring artifacts in the sinogram

Defective and mis-calibrated detector cells may cause artifacts in sinogram:



Ring artifacts

After the reconstruction process rings appear in the image domain:

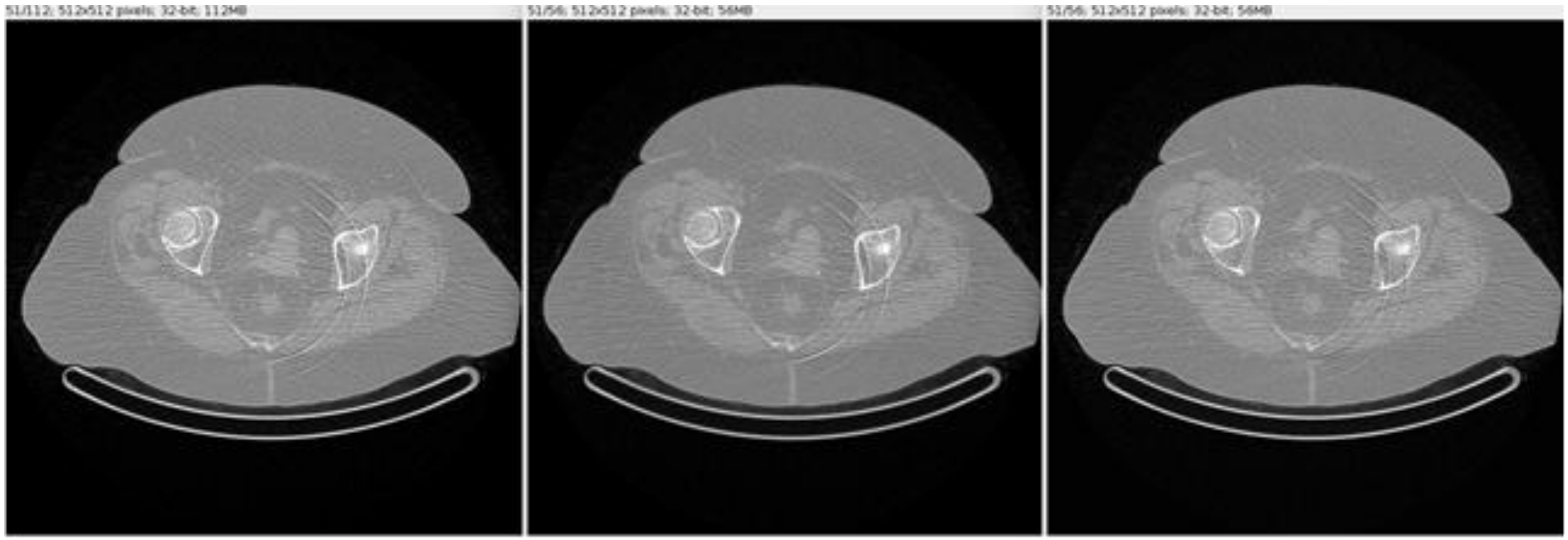


Classification of ring artifact correction methods

- Image-space methods
    - Reconstruct image and remove rings in image space.
- Fourier-space methods
    - Apply low pass filter in fourier space (during the FDK reconstruction process)
- Sinogram-space methods
    - Find defective pixels and correct in sinogram space.

Ring artifacts in image space

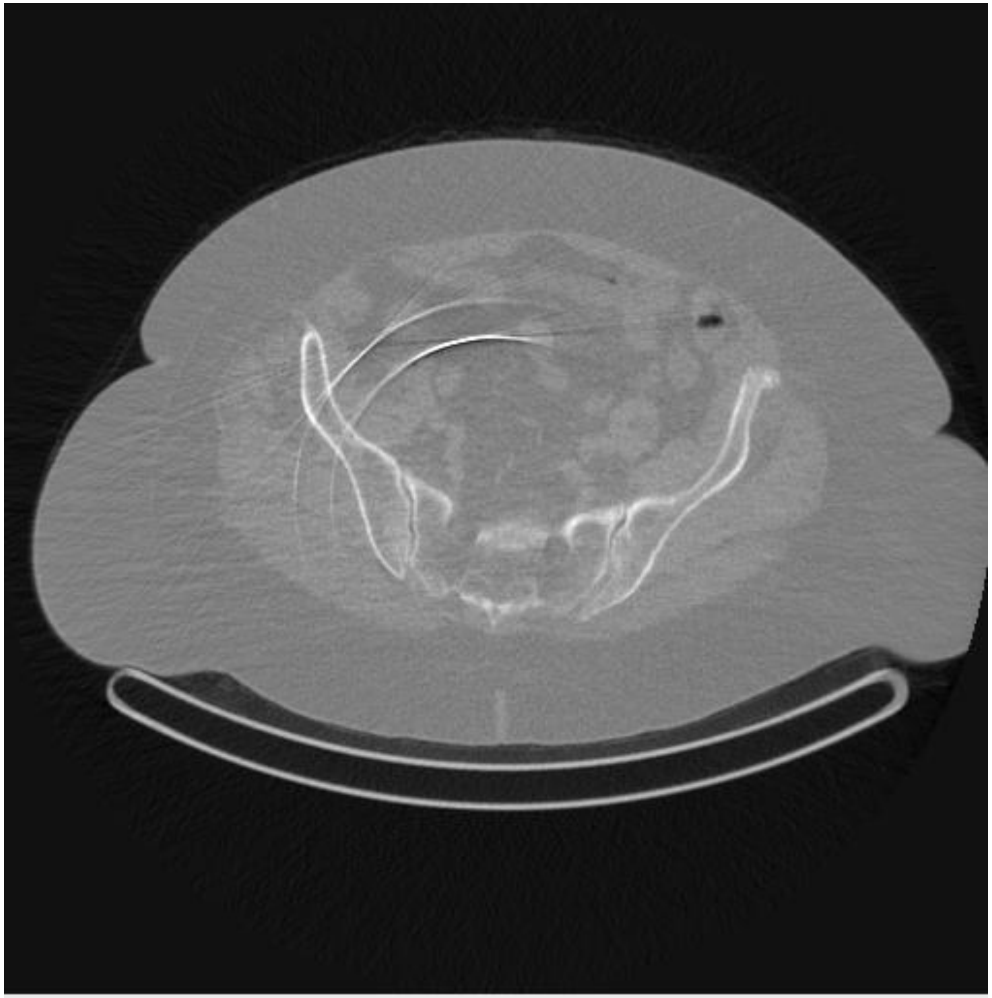
- Ring correction in polar coordinate (RCP)
    - Using polar coordinate transformation
- Ring correction using homogenity test (RCHT)

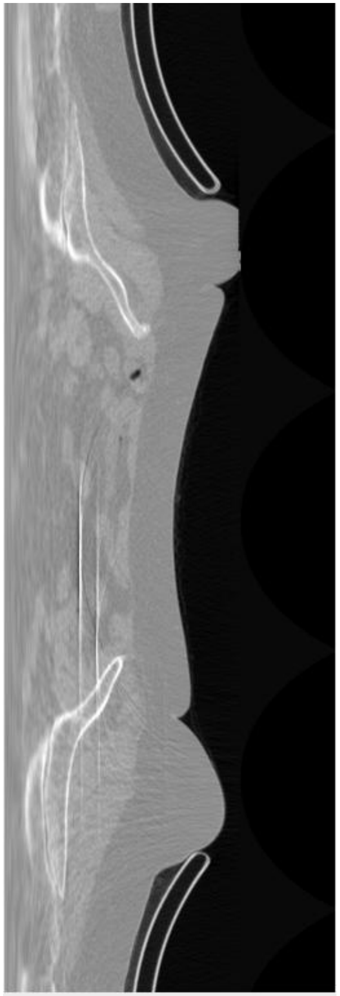


Original RCHT RCP

Ring artifacts in image space

The image based corrections use polar coordinate transformation





The filtered back projection in 2D:

Let $f : \mathbb{R}^2 \to \mathbb{R}$ be the original volume to be reconstructed.

The Lambert-Beers law determines the transmitted x-ray intensity I after traversing the volume f along a line L:

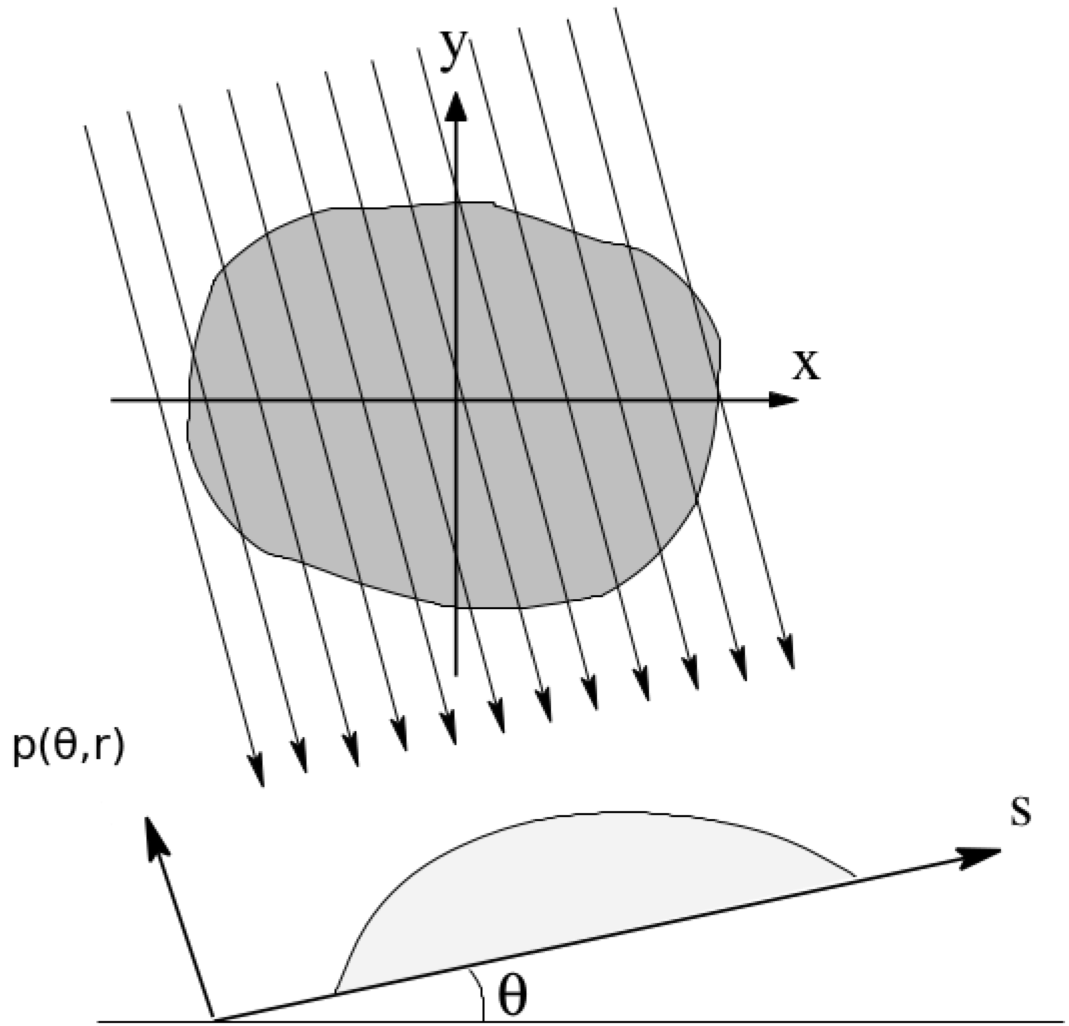
$$I = I_0 \cdot e^{-\int_{L_{\underline{x}}} f(\underline{x}) d\underline{x}}.$$

where I_0 is the known intenity of the radiation emitted at source. We get the sinogram which contains of the projections:

$$p(\theta, r) = \int_{L_{r,\theta}} f(r \cdot \cos\theta - s \cdot \sin\theta, r \cdot \sin\theta + s \cdot \cos\theta) ds$$

Reconstruction process

The filtered back projection in 2D:



Mediso

Reconstruction process

The filtered back projection in 2D:

From the projections we get the original volume to invert the Radon transformation:

$$f(x,y) = R^{-1}\{p(\theta,r)\}$$

The projection theorem:

Let P(k,θ) be the 1D Fourier transformation of the projections

$$P(k,\theta) = \int_{-\infty}^{\infty} p(\theta,r) \cdot e^{-2\pi i k r} dr$$

and let F(u,v) be the 2D Fourier transformation of f(x,y)

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y) \cdot e^{-2\pi i (xu+yv)} dx dy$$

Then $P(k,\theta) = F(k\cos\theta, k\sin\theta)$

Ring artifacts in fourier space

The Filtered Back projection uses Fourier transformation.

During the Fourier transformation we can use different filters and ring correction methods:

- Wavelet-Fourier (WF)
  - Apply low pass filter in Fourier space.
- Modified wavelet-plus-normalization (MWPN)

Ring artifacts in sinogram space

- Improved sinogram based deringing algorithm (ISDR)
- Ring detection with morphological operators (IMF)
- Median filtering algorithm (MedF)
- Median Filtering with Sum of Curve calculation (MSC)
- Iterative center weighted median filtering (ICWMF)

Deringing methods in sinogram space

ISDR selects defective cells into three sets:

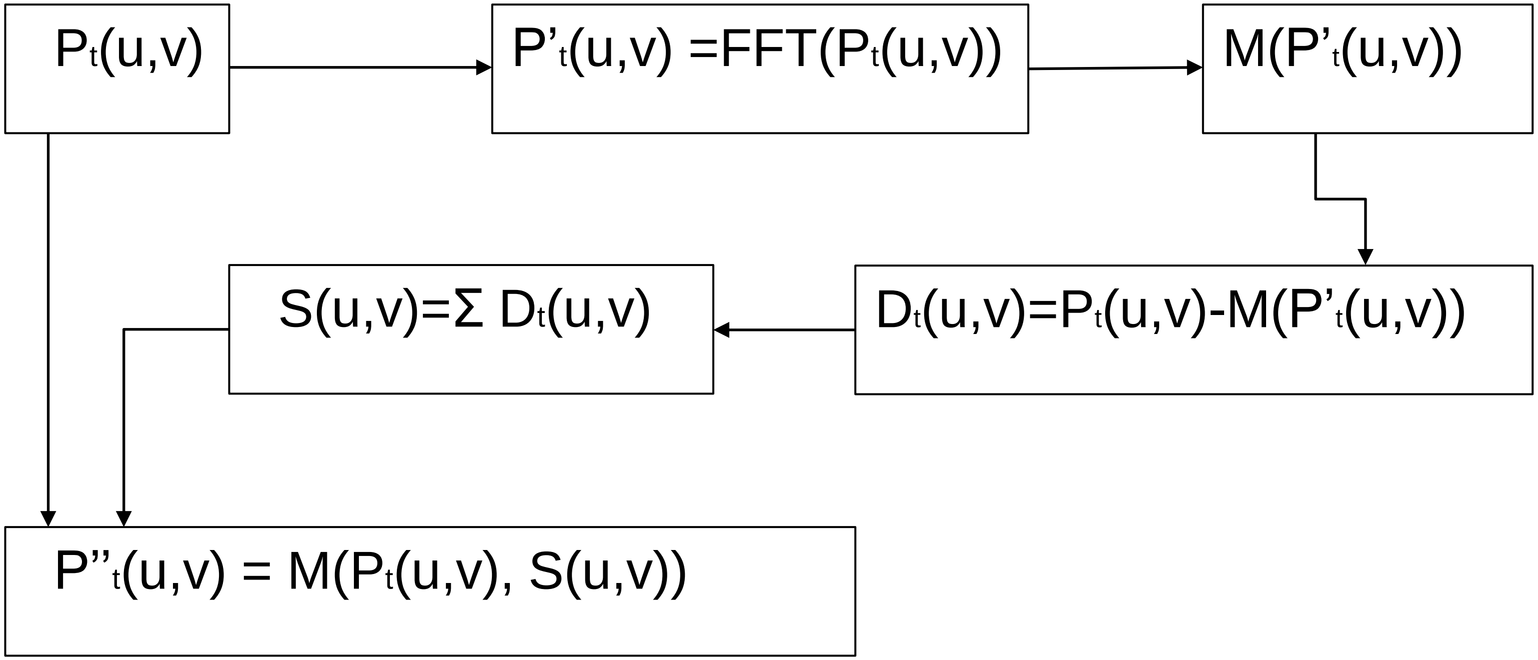
- Damaged cells (pixel value independent of neighboring data)
- Time independent mis-calibrated cells (fixed offset error)
- Time dependent mis-calibrated cells, (variable offset error)

IMF method uses mathematical morphological filters in three steps:

- IRE for remove intense rings.
- LRE for smoothing the mean cure.
- BRE for band ring elimination.

Deringing methods in sinogram space

MSC algorithm



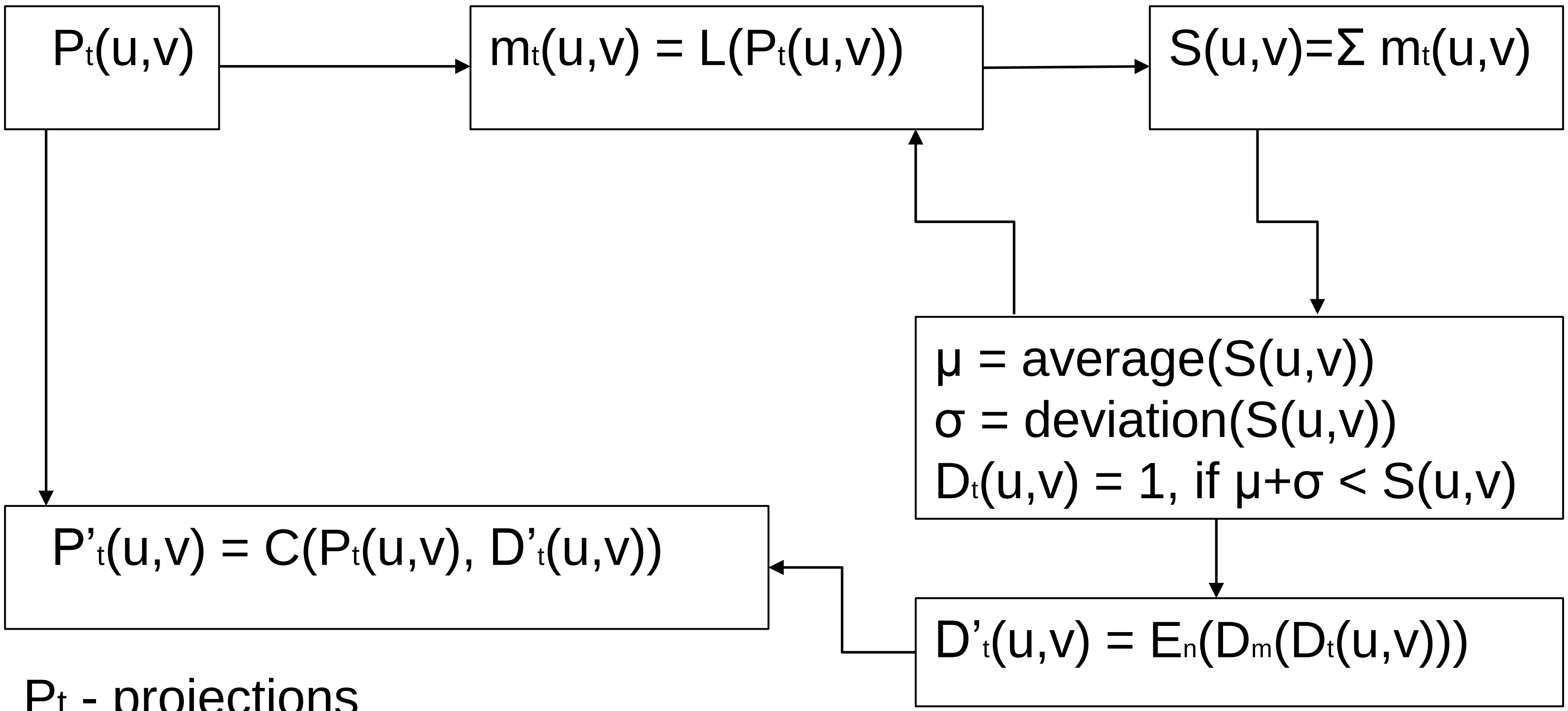
Pt - projections

FFT – Fast Fourier Transform

M – Median filter

Deringing methods in sinogram space

Our proposed algorithm



P_t - projections

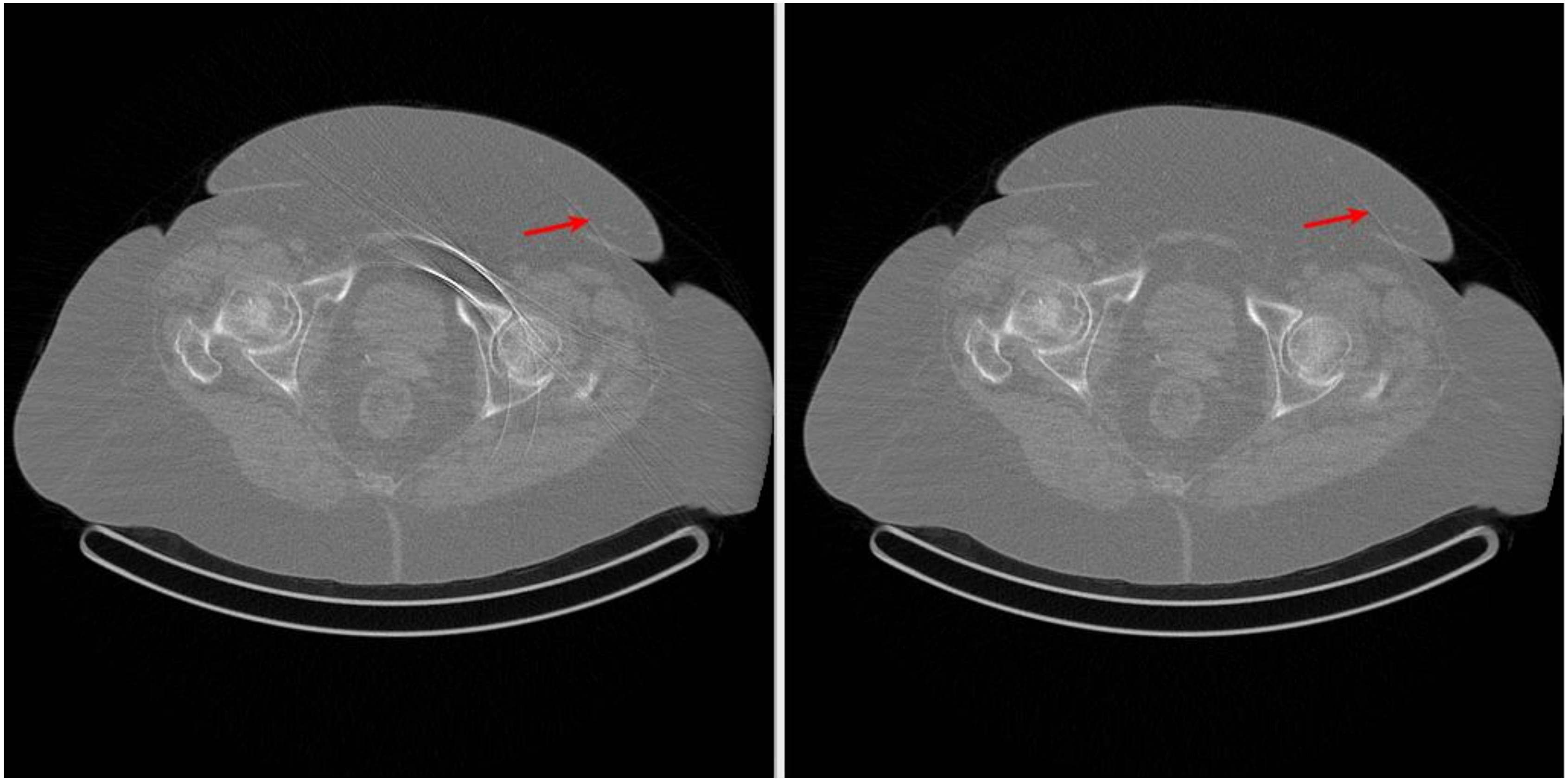
L – Local difference function

E_n – Erosion n times

D_m – Dilatation m times

Deringing methods in sinogram space

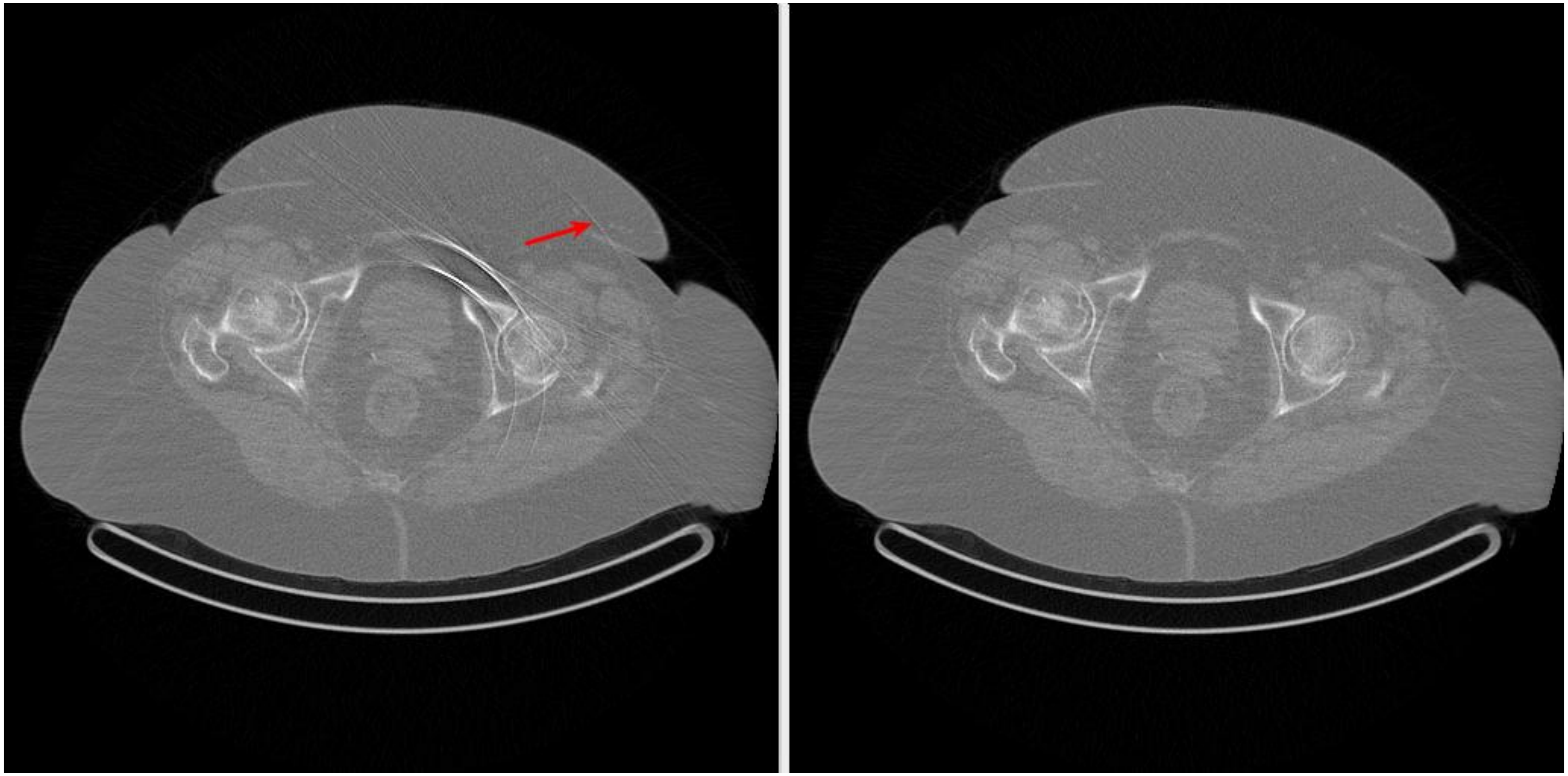
Comparison of original and MSC images



original MSC

Deringing methods in sinogram space

Comparison of original and our proposed algorithm



original

Our proposed algorithm

Implementation environment

CUDA (Compute Unified Device Architecture)

A parallel computing platform and programming model invented by NVIDIA. It increases in computing performance by exploiting the power of the graphics processing unit (GPU).

Hardware

Kepler microarchitecture

GeForce GTX TITAN Black

- compute capacity 3.5,
- memory 6 GB,
- CUDA cores 2688

Implementation environment

Using CUDA memory and texture

Store projections and reconstructed image in CUDA device memory:

```
cudaArray<float> projections;
```

Use texture to faster data access

```
texture<float, cudaTextureType3D> t_Projections;
```

Texture initialization:

```
t_Projections.addressMode[0] = cudaAddressModeClamp;
t_Projections.addressMode[1] = cudaAddressModeClamp;
t_Projections.filterMode = cudaFilterModePoint;
t_Projections.normalized = false;
```

Implementation environment

Texture usage

Set cudaArray pointer

```
cudaArray* deviceArrayPtr = projections->getArrayPtr();
```

Bind texture

```
cudaBindTextureToArray( &t_Projections, deviceArrayPtr,
&m_ChannelDesc );
```

Texture reference in kernel

```
float value = (float)tex3D(t_Projections, (float)x, (float)y, (int)z );
```

Unbind texture

```
cudaUnbindTexture( t_Projections );
```

Implementation environment

Speed comparison

Reconstruction of image with 1452 slices

Mode	Reconstruction time (sec)
Using texture	136.140972
Without texture	152.141553

Reason:

Border check is needed in the kernel without texture mode.

Thank you for your attention!

Mediso

www.mediso.com