Artifact removal in 3D Feldkamp Reconstruction with limited angles

Computational imaging project

Federico Augelli and Sandra Campacci

University of Bologna

June 19th, 2025

1. Introduction

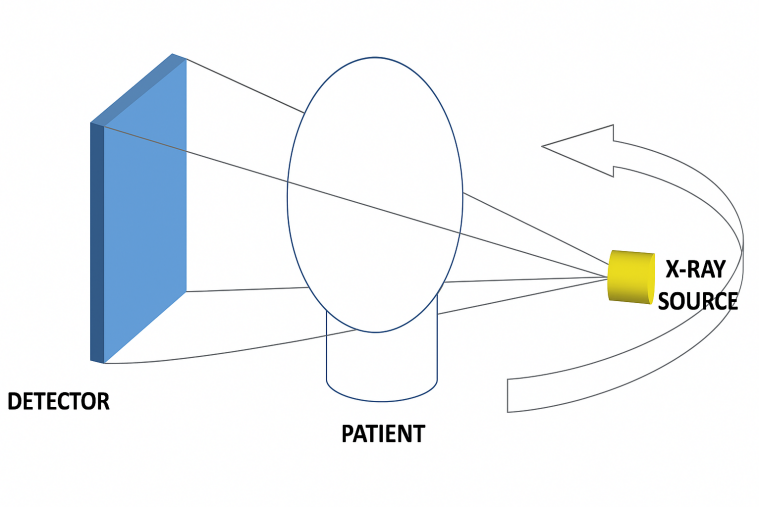
2. Construction of the 3D dataset

3. Projections and Feldkamp Algorithm

4. ResUNet 3D for denoise and results

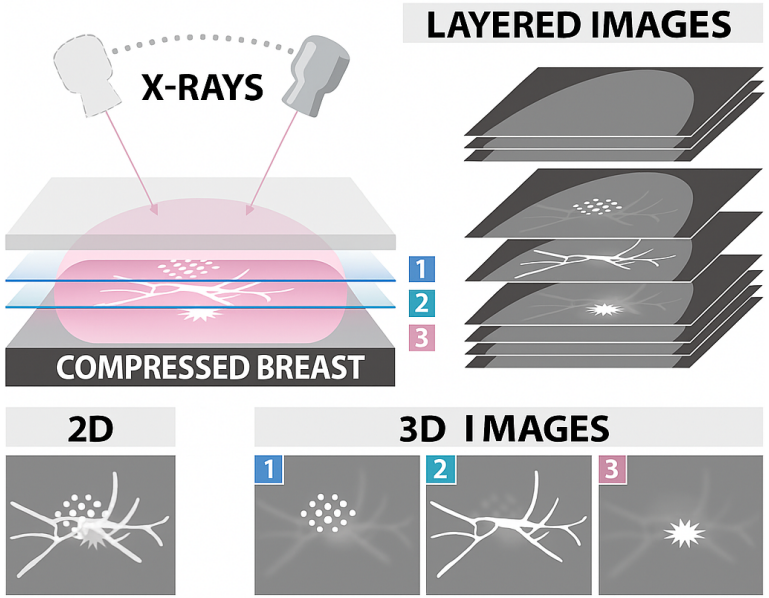
Cone Beam Computed Tomography

Cone Beam Computed Tomography is a medical imaging technique consisting of X-ray computed tomography where the X-rays are divergent, forming a cone.



Digital Breast Tomosynthesis

Digital Breast Tomosynthesis (DBT) may be seen as a 3D generalization of mammography. Starting from a limited number of projections, the goal of DBT is to provide a **volumetric reconstruction** of the scanned breast.



Aim of this project

The project consists of two main parts:

1. Create a **synthetic 3D dataset** containing objects that simulate the elements in the breast (carcinoma masses, microcalcifications and fibroglandular tissue's components);
2. Develop a framework using the ASTRA toolbox that performs **3D reconstruction from limited-angle projections** with the Feldkamp algorithm and subsequently removes artifacts through a **post-processing neural network**.

Aim of this project

The project consists of two main parts:

1. Create a **synthetic 3D dataset** containing objects that simulate the elements in the breast (carcinoma masses, microcalcifications and fibroglandular tissue's components);
2. Develop a framework using the ASTRA toolbox that performs **3D reconstruction from limited-angle projections** with the Feldkamp algorithm and subsequently removes artifacts through a **post-processing neural network**.

1. Introduction

2. Construction of the 3D dataset

3. Projections and Feldkamp Algorithm

4. ResUNet 3D for denoise and results

3D dataset

The dataset is composed of **100 volumes 256 x 256 x 50** which contain all the relevant anatomic components of interest during an examination.

Since the aim of breast imaging techniques is to detect microcalcifications and masses, each volume includes:

- Number of ellipses = `random.randint(1, 5) + 8`;
- Number of circles = `random.randint(1, 5) + 8`;
- Number of lines = `random.randint(1, 5) + 4`;
- Number of centered ellipses = `random.randint(1, N of ell)`, with the condition that if two ellipses overlaps, only the more opaque will be shown.

3D dataset

The dataset is composed of **100 volumes 256 x 256 x 50** which contain all the relevant anatomic components of interest during an examination.
Since the aim of breast imaging techniques is to detect microcalcifications and masses, each volume includes:

- Number of ellipses = `random.randint(1, 5) + 8`;
- Number of circles = `random.randint(1, 5) + 8`;
- Number of lines = `random.randint(1, 5) + 4`;
- Number of centered ellipses = `random.randint(1, N of ell)`, with the condition that if two ellipses overlaps, only the more opaque will be shown.

Example of volume in the dataset

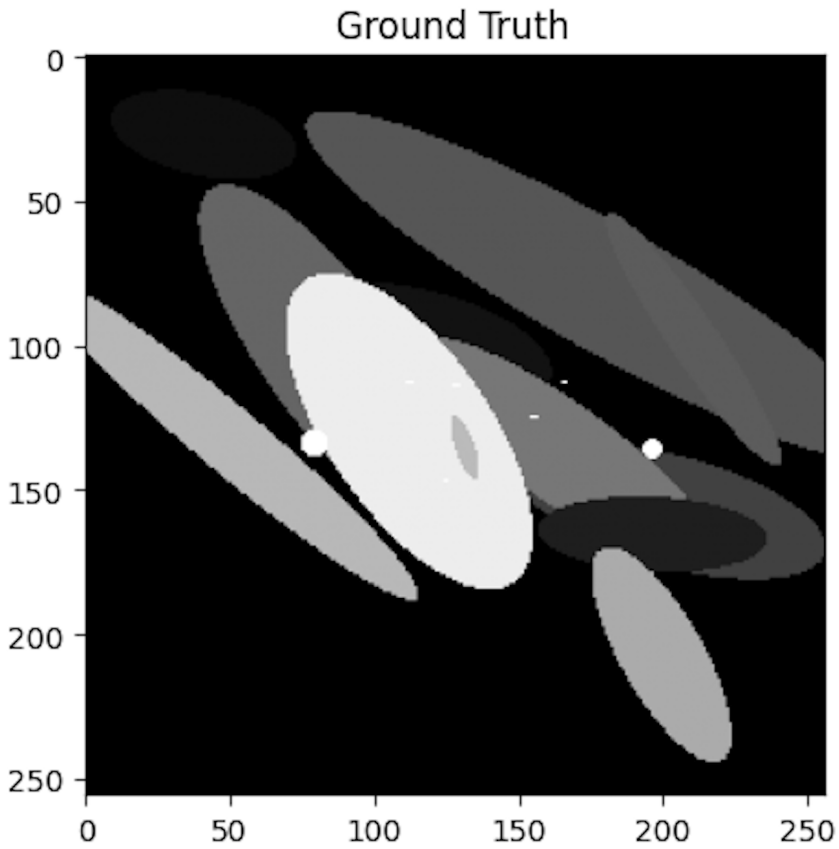


Figure: Slice of the 66th generated volume.

1. Introduction

2. Construction of the 3D dataset

3. Projections and Feldkamp Algorithm

4. ResUNet 3D for denoise and results

Limited-angle geometry in DBT

The setting for our DBT image reconstruction is:

- The **object volume** is discretized in 256 x 256 x 50 voxels;
- The position of the centre of the object is (-128, 128, -25);
- The **detector** is discretized in 500 x 700 pixels;
- The position of the centre of the detector is (-256, 128, -25);
- The pixel size is 0.95 mm x 0.95 mm;
- The **source** moves along a small circular arc, keeping the y axis fixed. We have tested the project using the following geometries (both with or without noise):
  1. **11 equally spaced angles between [-15,15]**;
  2. **11 equally spaced angles between [-8.5,8.5]**.
- The distance between the source and the axis of rotation of the object (SAD) is 800 mm.

Limited-angle geometry in DBT

The setting for our DBT image reconstruction is:

- The **object volume** is discretized in 256 x 256 x 50 voxels;
- The position of the centre of the object is (-128, 128, -25);
- The **detector** is discretized in 500 x 700 pixels;
- The position of the centre of the detector is (-256, 128, -25);
- The pixel size is 0.95 mm x 0.95 mm;
- The **source** moves along a small circular arc, keeping the y axis fixed. We have tested the project using the following geometries (both with or without noise):
  1. **11 equally spaced angles between [-15,15]**;
  2. **11 equally spaced angles between [-8.5,8.5]**.
- The distance between the source and the axis of rotation of the object (SAD) is 800 mm.

Limited-angle geometry in DBT

The setting for our DBT image reconstruction is:

- The **object volume** is discretized in 256 x 256 x 50 voxels;
- The position of the centre of the object is (-128, 128, -25);
- The **detector** is discretized in 500 x 700 pixels;
- The position of the centre of the detector is (-256, 128, -25);
- The pixel size is 0.95 mm x 0.95 mm;
- The **source** moves along a small circular arc, keeping the y axis fixed. We have tested the project using the following geometries (both with or without noise):
  1. **11 equally spaced angles between [-15,15]**;
  2. **11 equally spaced angles between [-8.5,8.5]**.
- The distance between the source and the axis of rotation of the object (SAD) is 800 mm.

Forward Projection

- A projection consists of the numerical measurements—specifically, the photon counts—captured by the detector;
- The values measured by the detector also contain the **noise** due to physical effects and inexactness of the detector counts;
- The noise is generally a mix of a Poisson and Gauss noise.

Forward Projection

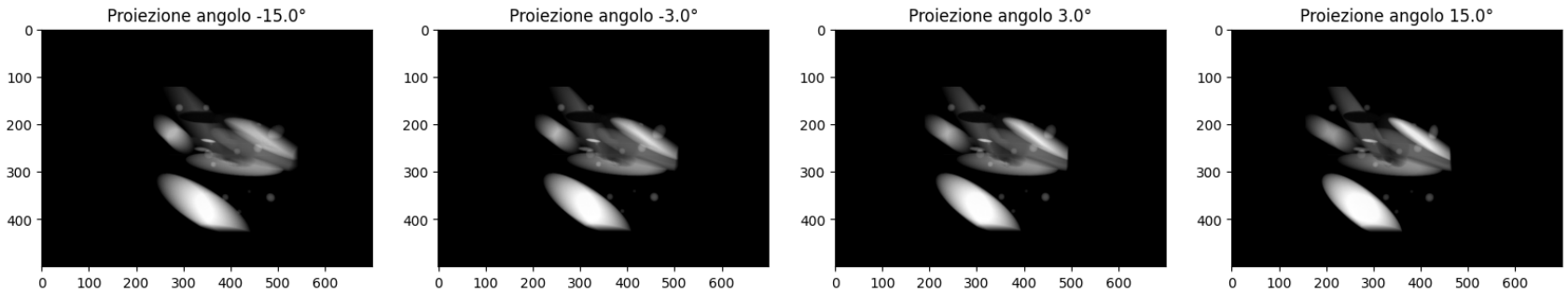


Figure: The projections of the volume 66 obtained with angles between [-15,15] without noise.

Forward Projection

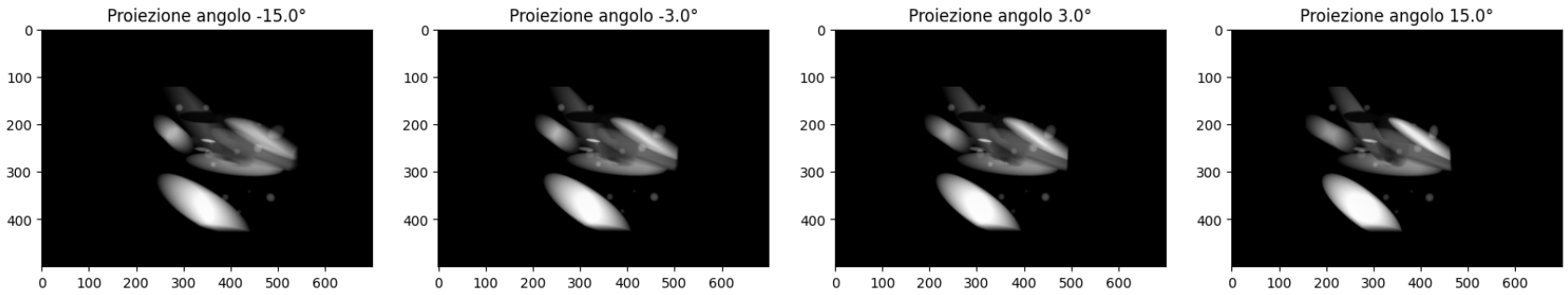


Figure: The projections of the volume 66 obtained with angles between [-15,15] with noise level 0.01.

Forward Projection

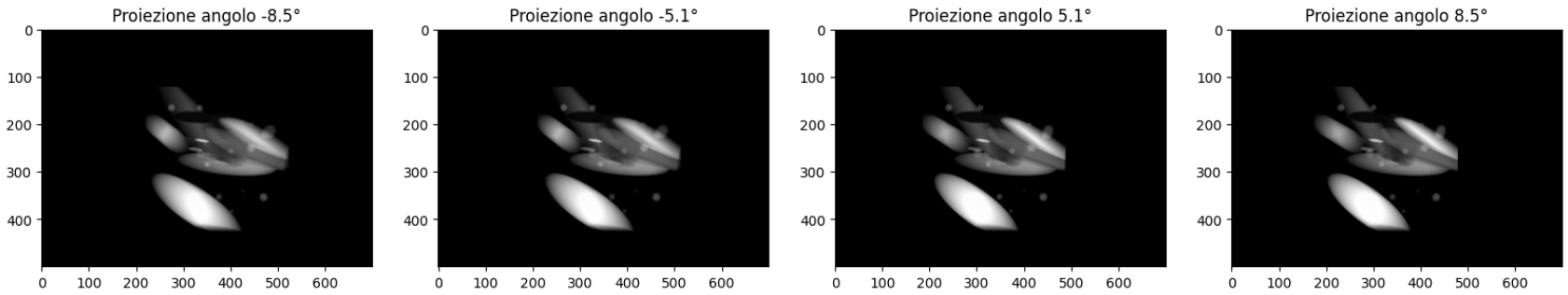


Figure: The projections of the volume 66 obtained with angles between [-8.5,8.5] without noise.

Forward Projection

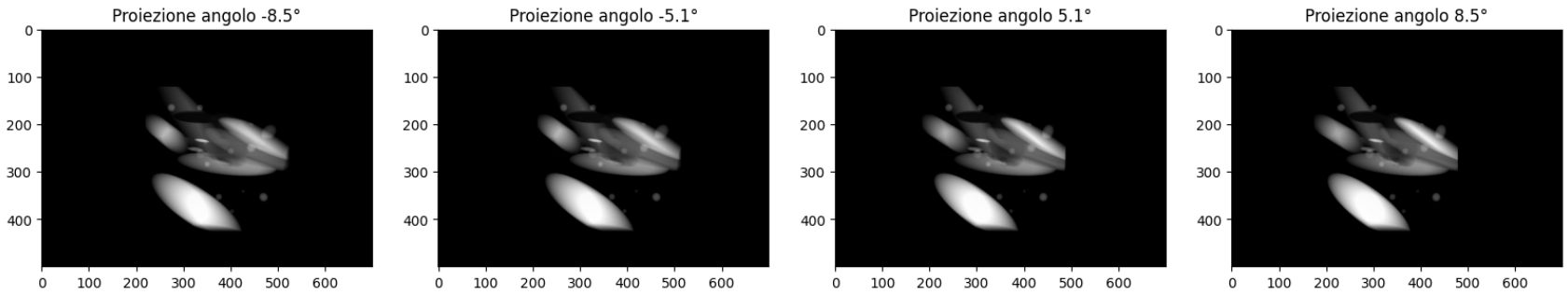


Figure: The projections of the volume 66 obtained with angles between [-8.5,8.5] with noise level 0.01.

Types of errors

For each of the executions in Feldkamp Algorithm, we use the following error metrics calculated between the reconstructed volumes and those of the dataset:

- **Relative Error (RE)** $= \frac{||x_{reconstructed} - x_{true}||}{||x_{true}||}$;
- **Peak Signal to Noise Ratio (PSNR)** $= 20 \cdot \log_{10}\left(\frac{\text{max pixel}}{\sqrt{MSE}}\right)$;
- **Structural Similarity Index Measure (SSIM)**, which focuses on structural information rather than just pixel-wise differences like the previous two.

Types of errors

For each of the executions in Feldkamp Algorithm, we use the following error metrics calculated between the reconstructed volumes and those of the dataset:

- **Relative Error (RE)** $= \frac{||x_{reconstructed} - x_{true}||}{||x_{true}||}$;
- **Peak Signal to Noise Ratio (PSNR)** $= 20 \cdot \log_{10}\left(\frac{\text{max pixel}}{\sqrt{MSE}}\right)$;
- **Structural Similarity Index Measure (SSIM)**, which focuses on structural information rather than just pixel-wise differences like the previous two.

Types of errors

For each of the executions in Feldkamp Algorithm, we use the following error metrics calculated between the reconstructed volumes and those of the dataset:

- **Relative Error (RE)** $= \frac{||x_{reconstructed} - x_{true}||}{||x_{true}||}$;
- **Peak Signal to Noise Ratio (PSNR)** $= 20 \cdot \log_{10}\left(\frac{\text{max pixel}}{\sqrt{MSE}}\right)$;
- **Structural Similarity Index Measure (SSIM)**, which focuses on structural information rather than just pixel-wise differences like the previous two.

Reconstructions with FDK

- The reconstruction of a volume consists in a stack of 2D images. Indeed we are working in a discrete setting, since the projections are computed from a finite number of angles;
- The reconstruction is based on the idea of back projecting onto the voxels of the object the values measured for each pixels of the detector at each scanning angle.

Results and comparisons

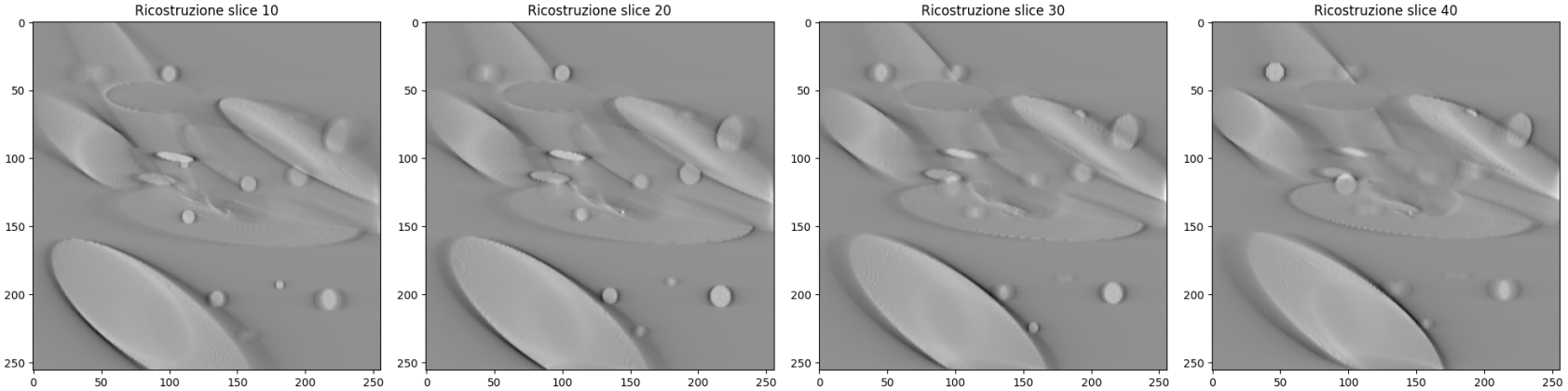


Figure: The reconstruction of the volume 66 from the projections obtained with angles between [-15,15] without noise.

relative error	PSNR	SSIM	RMSE
0.826784	11.17382	0.201425	0.287535

Results and comparisons

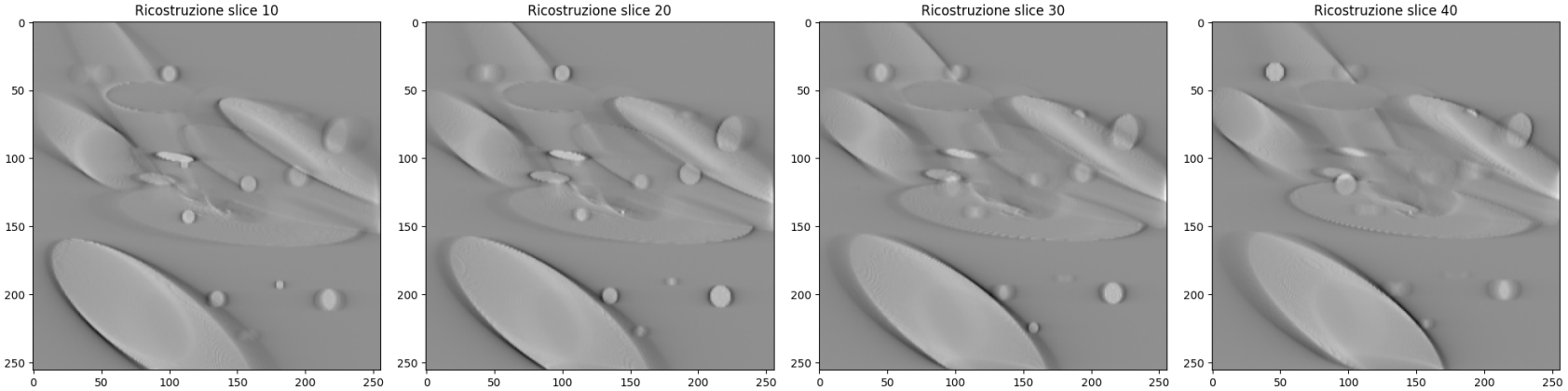


Figure: The reconstruction of the volume 66 from the projections obtained with angles between [-15,15] with noise level 0.01.

relative error	PSNR	SSIM	RMSE
0.835754	11.005342	0.196439	0.289912

Results and comparisons

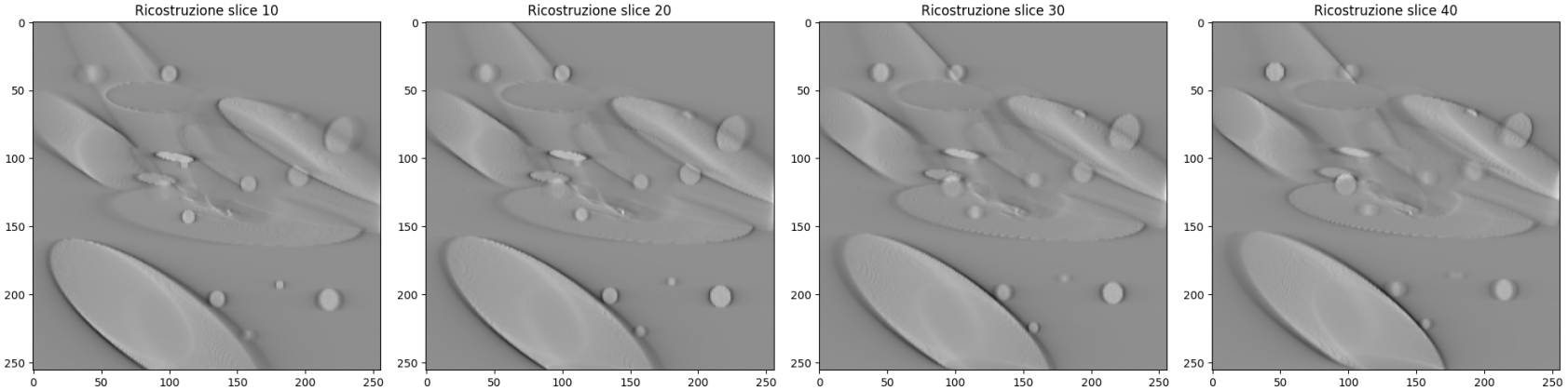


Figure: The reconstruction of the volume 66 from the projections obtained with angles between [-8.5,8.5] without noise.

relative error	PSNR	SSIM	RMSE
0.857398	10.69053	0.194472	0.299231

Results and comparisons

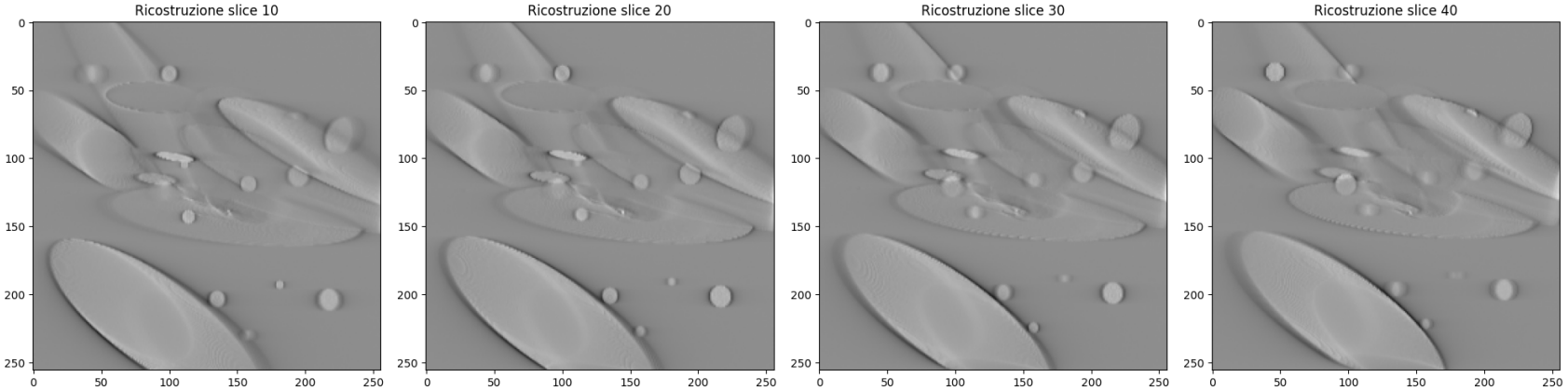


Figure: The reconstruction of the volume 66 from the projections obtained with angles between [-8.5,8.5] with noise level 0.01.

relative error	PSNR	SSIM	RMSE
0.860429	10.56624	0.194701	0.299721

Average of the errors

angles	noise	relative error	PSNR	SSIM	RMSE
[-15,15]	0.00	0.832946	11.02590	0.200415	0.289848
[-15,15]	0.01	0.833177	11.02329	0.195927	0.289937
[-8.5,8.5]	0.00	0.859352	10.74269	0.195864	0.299014
[-8.5,8.5]	0.01	0.859558	10.74059	0.192215	0.299088

1. Introduction

2. Construction of the 3D dataset

3. Projections and Feldkamp Algorithm

4. ResUNet 3D for denoise and results

Neural Networks

- Neural networks can automatically learn priors from large datasets, capturing complex image features without manual tuning;
- On the other side we have a data dependency, since large and high-quality datasets (ground truth) are required for training.

ResUNet architecture

- The denoiser can be implemented by a **Residual Unet 3D** that is trained from **Feldkamp reconstructed volumes** to **ground truth**;
- The ResUNet is a type of Convolutional Neural Network with an Encoder-Decoder structure and residual blocks. It was trained on 400 pairs as training set and 100 as test set;
- Volumes are split into **5 sub-volumes of 10 slices each**.

ResUNet architecture

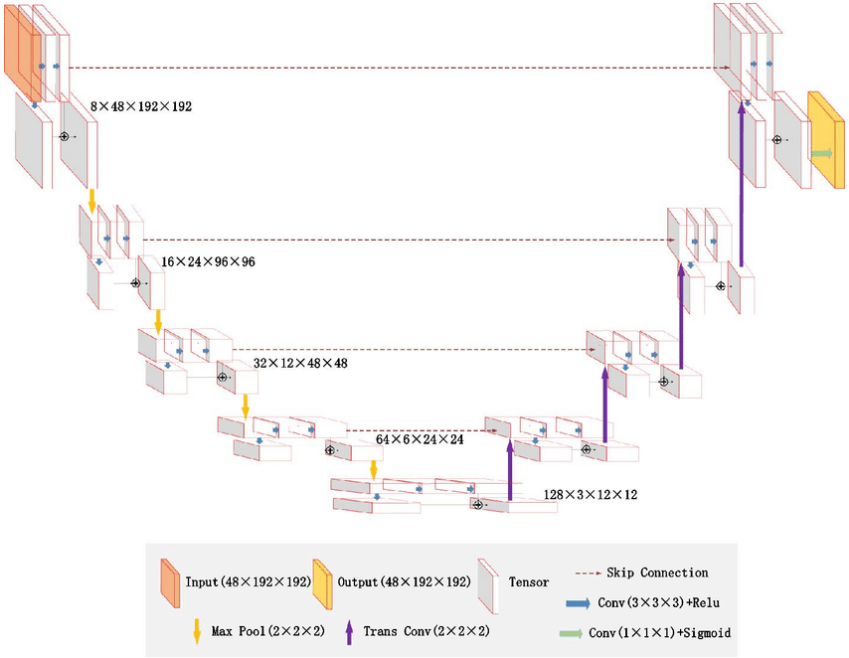


Figure: ResUNet architecture.

Loss function

Several **loss** function were used:

- The first loss function was compose of **RMSE** and **PSNR**. It had low quality but high performance in SSIM and PSNR metrics;
- The second iteration of the loss function has two new metrics: **Frequency Domain Loss** and **Edge Preservation Loss**.

Average of errors

	PSNR	SSIM	loss	val loss	epochs	time
train 1	30.20	0.7313	0.1598	0.2540	20	11m
train 2	28.39	0.7119	0.1470	0.2767	50	36m
train 3	BAD	BAD	BAD	BAD	20	11m
train 4	25.38	0.5622	10.8865	13.1988	20	11m
train 5	5.13	0.0488	8.2243	12.5267	50	35m
train 6	23	0.5	9.2875	11.5997	20	11m
train 7	BAD	BAD	BAD	BAD	20	11m

Results on noisy projections

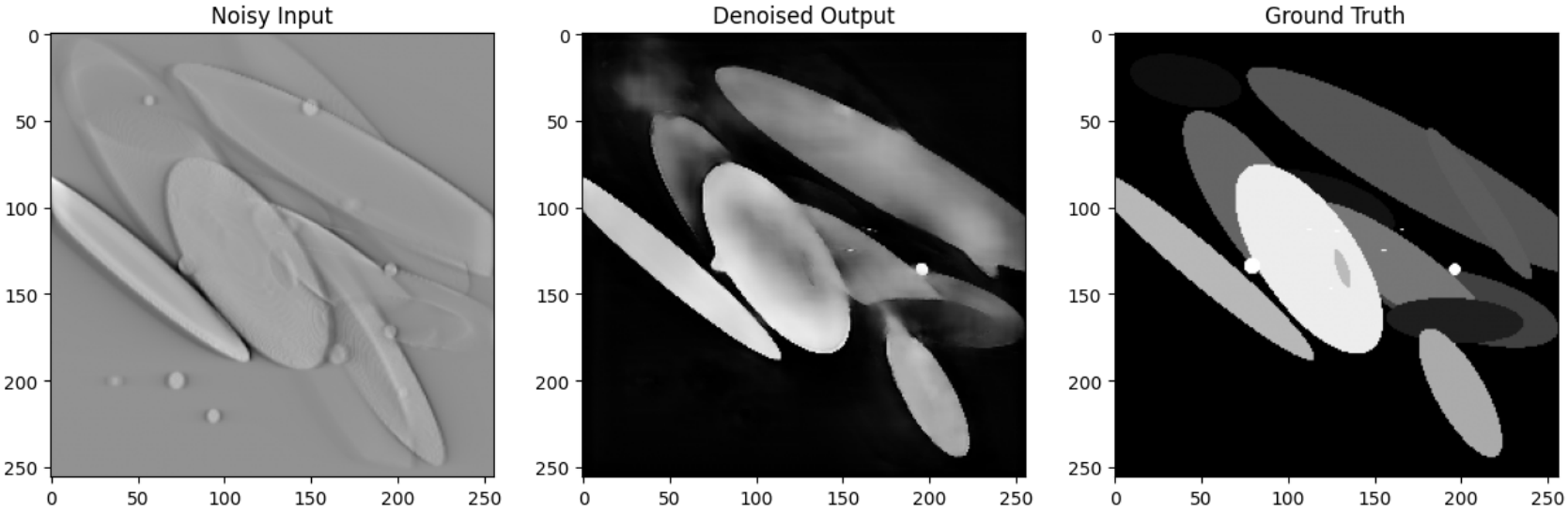


Figure: The denoised image from noisy reconstructed volume.

Average of errors

	psnr	ssim
Reconstruction	11.02	0.2004
Denoised	25.38	0.5622

THANK YOU FOR YOUR ATTENTION!!