Signals and Systems | **Introduction 2** | **WS 18/19**
Bildgebende Systeme 1 | | **Version: 9.11.15**

Creating a sinogram

Goal:

In this exercise sheet you will implement a forward projection in order to simulate a simple 2D CT scanner with parallel geometry. Again, this exercise is voluntary and you don't have to submit anything.

Introduction:

In a traditional X-ray system which consists of a stationary X-ray source and a stationary detector, the measurement process is equivalent to *projecting* the attenuation values of a 3D object (e.g. a patient) onto a 2D detector. The result is a simple 2D X-ray image everybody should be familiar with.

In contrast, in most modern CT machines only a very thin slice of the object is illuminated by X-ray radiation. This means that the central part of a simple CT simulation can be understood as the projection of a 2D image on a 1D detector. This will be explained in part I.

The second important part in a CT system is the fact that not only one, but *many* projection images from all around the object are taken. As you will see in part II, in our implementation the object is rotated, while the source/detector combination is stationary. As you may have learned in the lecture, a clinical CT works the other way around: the patient is fixed while the source and the detector rotate on a circular trajectory. On the other hand, other systems (e.g. Micro-CTs for material testing) use a stepper motor to rotate the object.

Part I: A simple projection

1) Create a software phantom with the Matlab command `phantom(200)`, save it to a variable of your choice and display it in grayscale via `imagesc()` and `colormap(gray(256))`.

2) Use the `sum()` instruction to "project" the image along its columns onto a virtual detector, which lies on the bottom of the image. Hint: `sum()` lets you specify the axis along it will sum the values. If you plot the result it should look like the one in Figure 1.

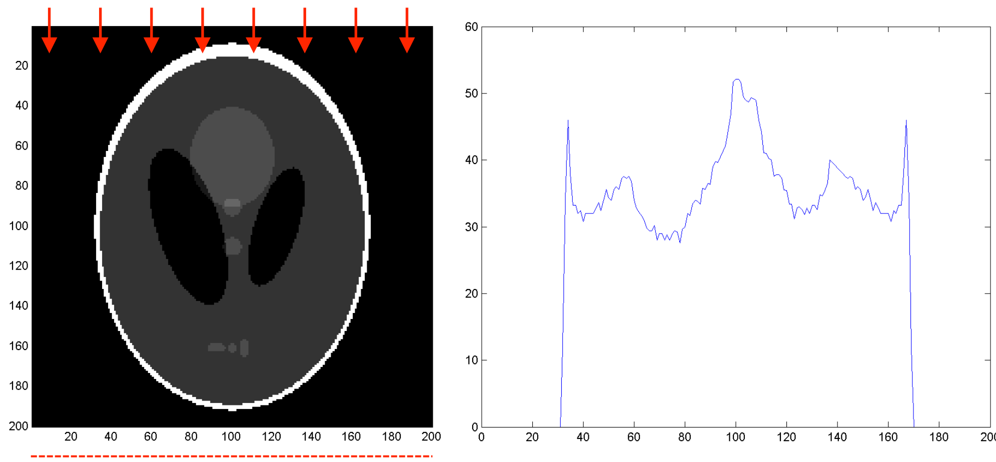

Figure 1: Left: The software phantom. The arrows indicate the projection direction and the dotted line on the bottom is the virtual detector onto which the values are projected. Right: The projection data.

Part II: A complete CT measurement

In a first step you should play with the Matlab command `imrotate()`. Use it to rotate the phantom from part I and display the results for different angles. If you use `imrotate()` without any special parameters you will see that the rotated image has a different size than the original one. This will lead to problems later on so you should understand and use the parameter `crop`.

After you have understood how `imrotate()` works you should now create a file called projection.m which contains a function of the same name. The function header should look like this:

```
function sinogram = projection(image)
```

The task of this function is two-fold: i) rotate the image between 0 and 179 degree and ii) for each angle (or “view”) calculate the projection data and save it as a sinogram.

In more detail, in this function you should write a main loop from 0:179. For each angle:

a. rotate the image around this angle (don’t forget `crop`)

b. sum the rotated image along its first dimension (e.g. along the columns rows)

c. collect this data into a new 2D matrix. A single row of this matrix is the result of step b)

In order to verify your function, go to the moodle-page of this course, download the image `CTLab-Introduction2.jpg` and save it to your working folder. Use the Matlab function `imread()` to load it. Since it is a normal color image it has the size (n*n*3), where 3 is the number of color channels. Use only the first channel for this exercise.

If you have done everything right, the resulting sinogram should look like the one in Figure 2.

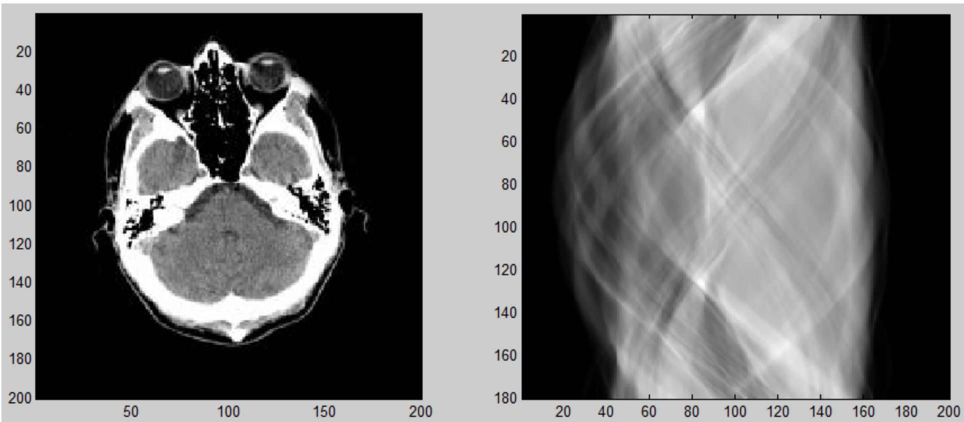

Figure 2: An example image (left) and its Radon transform (right)

Part III: Origin of the term *sinogram*

As the name suggest, it has something to do with the sinus function. To understand it, create an empty image with `zeros()` and set some single pixels to 1. If you then calculate the forward projection of this nearly empty image you will see several sinusoidal curves in the sinogram. This means, that the contribution of a single pixel in the image is distributed along a sinusoidal curve in the image. Therefore, the complete sinogram is the combination of all sinusoidal traces from all the pixels in one image.

It is important to note that bright pixels further away from the center of the image generate sinus curves with larger amplitude.