

X-RAY COMPUTED TOMOGRAPHY

Bc. Jan Kratochvíla

Czech Technical University in Prague
Faculty of Nuclear Sciences and Physical Engineering

Abstract

Computed tomography is a powerful tool for imaging the inner structure of various objects. It is based on penetrating radiation (such as X-Rays) which passes through an object and is attenuated by the material of the object. If the object is rotated and the passed radiation is detected at many angles, a cross-section of the object can be reconstructed using computers. Computed tomography was invented in 1972 by Hounsfield and because of its non-destructive nature, it has found its widest application in medicine for diagnosing various diseases.

In this project, MATLAB was used as the key tool in building a device for X-Ray computed tomography. The system consists of an X-Ray radiation source, a rotation stage used for rotating the object under investigation, the radiation detector, and a computer. MATLAB controls the rotation stage by means of ActiveX drivers which are included in the device. After each step, an image is acquired by an external program that is called from MATLAB (using *dos* command). When the object has been rotated by 180 or 360 degrees, the cross-section of the body can be computed. This is done by MATLAB routines *fan2para* and *iradon* (both are part of the Image Processing Toolbox which comes as a part of MATLAB). Before calculating the resulting images, filtering of the individual images from the noise and various errors is essential.

The system has been successfully built and provided satisfactory results. Further work is expected to enhance the quality and resolution of the images.

1 Block scheme and function overview

The system built was an instrument for performing X-Ray computed tomography. It consists of four main parts:

- X-ray source (tube)
- rotation stage
- X-ray detector
- computer

First, images of the object under investigation must be acquired. This is done at different angles to cover the full cycle, i.e. 360 degrees. The object is rotated by the rotation stage about 1 degree per a step.

The next step is the mathematical calculations. The goal is to get a cross-section of the object at a given height. After enhancing the quality of the acquired images (offset and open-beam corrections), sinograms are calculated. A sinogram is a new image which consists of all 360 previously acquired images at a particular height. Let's say we want to obtain a cross-section at a height corresponding to 100th pixel of the images. We take the 100th line of each image and put it gradually after each other in the resulting sinogram. For each slice (cross-section) of the object, one sinogram is necessary. Having computed the sinograms, we can arrive at the the desired cross-sections of the object using MATLAB function *iradon*.

2 Used devices

Let us briefly describe devices used in our setup.

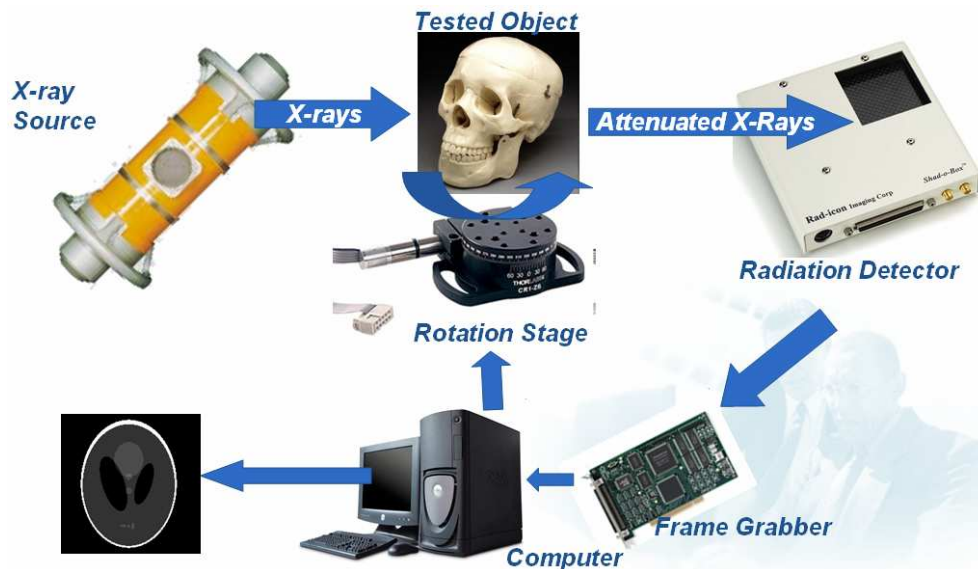


Figure 1: Block scheme of the system

2.1 Shad-o-Box X-ray camera

Shad-o-Box X-ray camera was used to detect the X-Ray radiation. It is based on a CMOS photo diode sensor which has 10 lines per millimeter resolution in a 2000 by 2048 pixel array. The sensor is produced for two energy ranges (10-50 kV and 10-160 kV). We used the extended version for up to 160 kV. The digital output is 12 bits. The area of the sensor covers a square with a side of 10 cm.

2.2 PXD1000 frame grabber

The X-Ray camera was connected to a digital frame grabber which is a device that acquires the images from the sensor and provides the data to a computer through the computer's PCI bus. The PXD1000 frame grabber was used.

The frame grabber was provided with libraries that enabled us to develop custom applications in C programming language.

2.3 Rotation stage

For rotating a sample, a rotation stage by Thorlabs was used. It is a worm driven device which is controlled by T-Cube DC Servo Controller (TDC001) by the same company. The controller is connected to a computer by USB interface. It has its own software called APT User where the motion can be easily controlled. It also provides ActiveX components which provide programmable interfaces that can be incorporated into client applications (e.g. in MATLAB).

2.4 X-Ray tube

Since there was not a separate X-Ray tube available which we could use only for the purposes of our device, we used the X-Ray tube from tomographic scanner BT-400 which is installed in our department. Even though the maximum voltage of the Isovolt 420/5 tube is 420 kV, we could only use 160 kV maximum because the X-Ray camera range was limited by this energy.

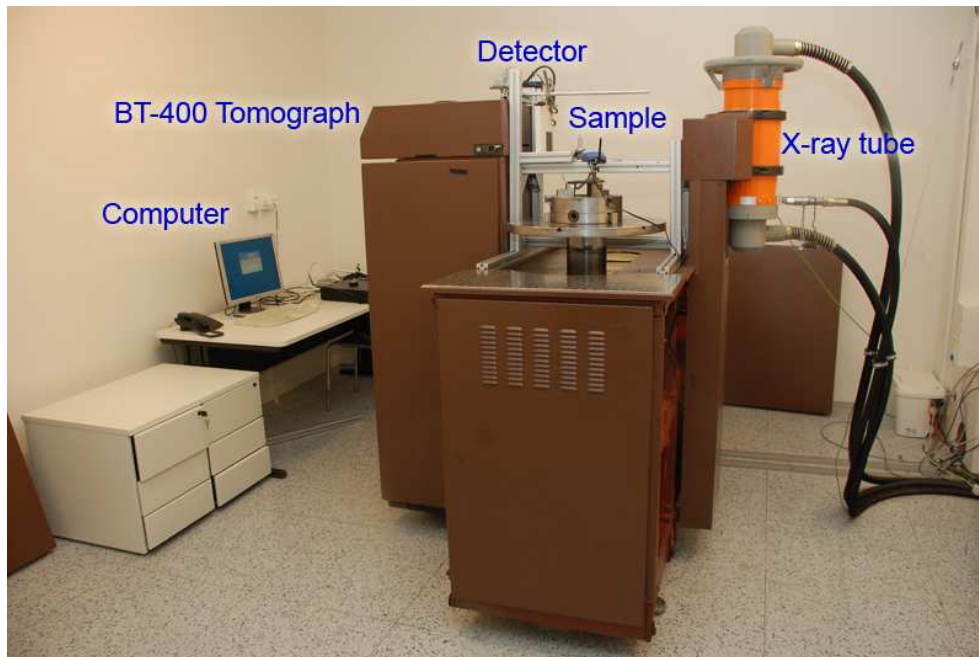


Figure 2: Setup of the whole system

3 Images acquisition

3.1 Sample rotation

The principle of the computed tomography is to take projections of the sample at many different angles. To achieve this, either the sample or the source and the detector are rotated. In our setup, the sample was rotated using the rotation stage CR1/M-Z6 together with the servo controller TDC001 by Thorlabs company. The controller was connected to a computer using the USB interface. We could not use the supplied software for controlling the motion because we needed to acquire the images after each step. Therefore, we used the ActiveX components to control the motion of the rotation stage directly from MATLAB.

3.2 Image capturing

The Shad-o-Box X-Ray camera was supplied with a Windows software called ShadoCam. It provides a simple user interface to capture, display, and save images from the camera.

However, the program needs an interaction with a user. There is neither a command line mode nor a batch mode. For our purposes it was necessary to be able to acquire a large number of images without any user interaction. Thus, ShadoCam software was not suitable for this task. It was necessary to code a simple program because there was not any other utility supplied to do this. To control the frame grabber, we used the libraries supplied with the frame grabber.

3.3 Correct exposure

In order to get well exposed images which are not saturated or all black, the correct exposure has to be determined so that the image captures all details of the sample. The exposure consists of three adjustable parameters:

- integration time (set on the flat panel)
- voltage (set on the X-ray tube)
- current (set on the X-ray tube)

Both voltage and current are set on the X-Ray tube control panel and influence the characteristics of the X-Ray radiation emitted. The number of emitted photons is influenced by the current and it corresponds to the brightness of the image (together with the integration time). The voltage controls X-Ray penetration as it determines the energy range of the radiation. It can be related to the contrast of the image.

3.4 Image corrections

The Shad-o-Box X-Ray detector which we used is based on CMOS technology. As opposed to CCD, CMOS active pixel sensors are less linear in their response. To decrease the influence of the non-linearity effects on the resulting picture, it is necessary to employ pixel corrections methods. The standard gain correction consists of two steps – a dark offset calibration and gain correction (also referred to as flat-field or open-beam correction). The first one compensates for the readout noise, ADC offset, and dark current.

The dark offset image is taken before or after tomography measurements with no radiation applied and with the same integration time as the one which will be used during the measurements. The dark offset image is simply subtracted from each acquired image.

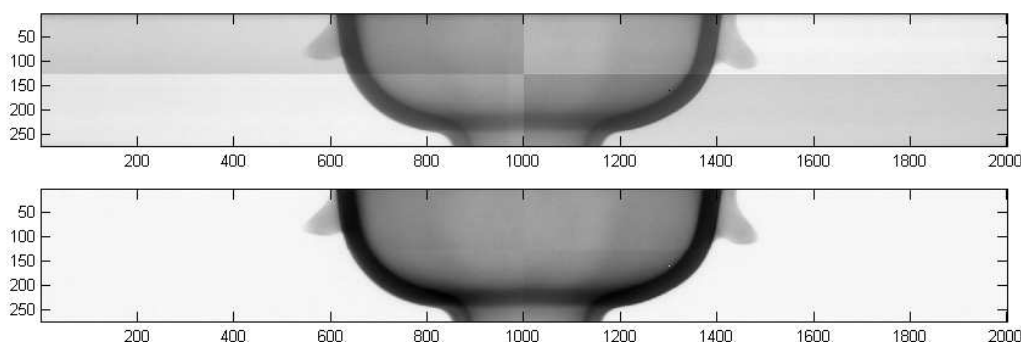


Figure 3: Comparison of a corrected image to the original

The flat-field images are obtained at some arbitrary input signal level (usually around 50% of the saturation signal but for best results, it should be similar to the signal levels expected in the final images). Each final image is divided by the flat-field image and multiplied by the average value of the flat-field image.

Since every final image is processed using the dark offset and flat-field image, every imperfection or error in those images will show in all the images (resulting in ring artefacts, for example). Thus, it is very advisable to take many calibration images (10 to 20) and then calculate their average to reduce noise and statistical errors.

3.5 Spot filtering

There are some pixels in the detector which are defective – their output level is either saturated, dark or very different from all the surrounding pixels. Such pixels cause ring artefacts in the final reconstruction.

We have implemented a spot filter to filter out these pixels. Since it consumes a lot of computation time, we used it to filter only the flat-field images which leads to good results in removing the ring artefacts. The spot filter is a modified median filter. Whereas the median filter acts on all pixels and replaces them with the median value of pixels in some surroundings (typically 3×3), we implemented a thresholded median filter.

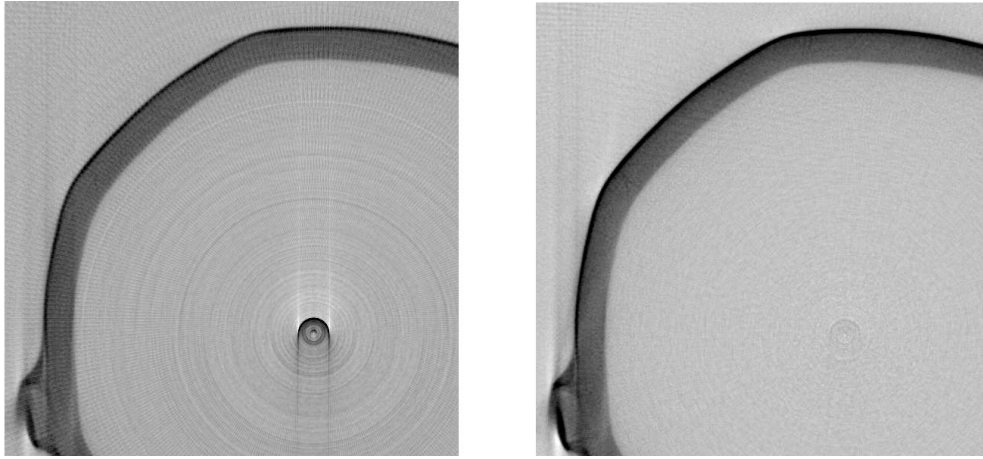


Figure 4: Ring artefacts: reconstruction without and with spot-filtered flat-field correction

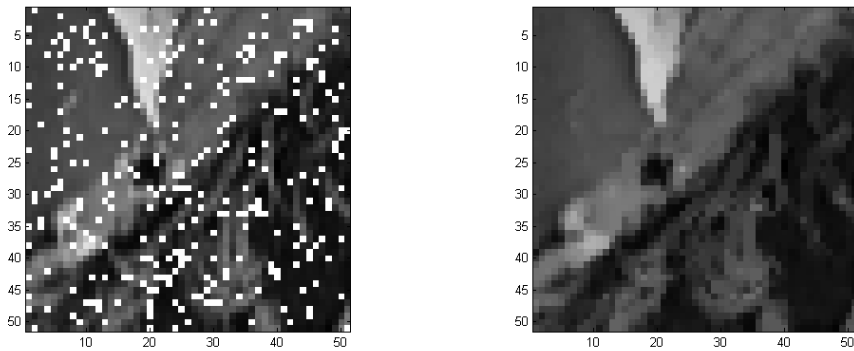


Figure 5: Images: original image and spot-filtered image

4 Image reconstruction

After images over the whole cycle (360 degrees) have been acquired, the image reconstruction can be carried out. The reconstruction is based on the Radon transform and filtered back-projection. It uses MATLAB script *iradon* which is a part of basic MATLAB distribution.

4.1 Sinograms

A set of projection data for a given slice through an object is called a sinogram. It is visualized as an image where the number of lines corresponds to the number of projections taken. Each line then represents a projection at a given angle. The number of columns is equal to the width of the detector. The reconstruction of a slice is performed from the data recorded in a sinogram.

For each desired cross-section of the object, the sinogram must be built. It is done by taking the appropriate line from each acquired image (corresponding to all angles covering the full cycle) and putting it gradually after each other.

4.2 Fan beam geometry

Before *iradon* can be applied, the conversion of the measured data from the fan beam geometry to the parallel beam geometry. The reason is the input requirements of the *iradon* function. It calculates the reconstructions from the parallel beam projections whereas we acquire the data using the fan beam because the source of the radiation is a small spot of the X-ray tube. To convert the sinogram which we acquired using the fan beam, script *fan2para* was used. It is a part of the Image Processing Toolbox of MATLAB.

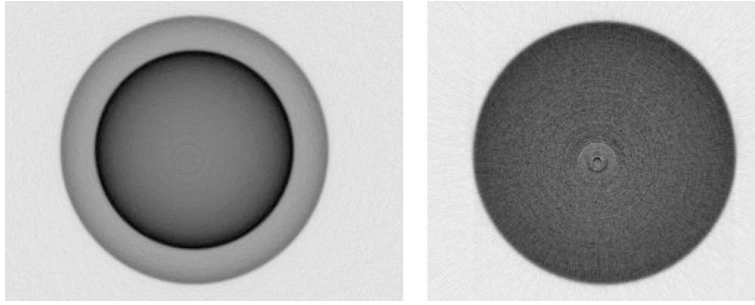


Figure 6: Reconstruction difference between not converted data (left) and fan beam data transformed to the parallel beam data (right)

4.3 Center of rotation calculation

The next step is to determine the center of rotation position in the images. It is essential that the center of rotation of the sample is perfectly aligned with the center of the detector (image). Best effort is made to align the physical geometry of the system so that the rotation stage axis is in the middle of the detector. However, it is hard to do with a pixel accuracy. It is better to use an algorithm which calculates the center of rotation from the measured data.

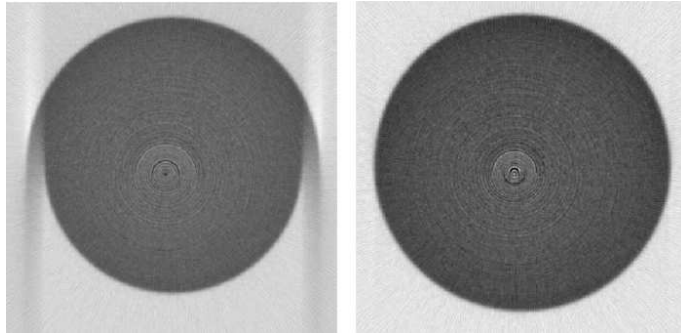


Figure 7: The influence of the center of rotation position. On the left: center of rotation wrongly placed by 12 pixels. On the right: Correct position of the center of rotation.

5 Filtered back-projection

After all three previous preparatory steps have been performed, the sinogram is passed to the *iradon* function which calculates the final reconstruction. The algorithm is as follows:

1. **Projections are filtered.** This is an essential step as the reconstructions without the filtering would appear blurred. There is a wide range of filters to choose from: Ram-Lak, Shepp-Logan, Hamming, Hann, and none. They differ in speed and efficiency. The ideal filter is a ramp filter which is not physically viable so its cropped version (Ram-Lak) is used. As the Ram-Lak filter is sensitive to noise in projections, we used the Hamming filter which multiplies the Ram-Lak filter by a Hamming window.
2. **Backprojection.** The filtered projections are backprojected to obtain the final reconstruction. It is done in a loop over the angles of rotation. Interpolation is necessary during the backprojection; we used linear interpolation because of the speed.

6 Tomography measurements procedure

In this section, we are going to describe the overall procedure to get a final image (slice, cross-section) of the object.

1. **Put the object on the rotation stage.**
2. **Initialize the rotation stage control.**
3. **Set the correct exposition:** Turn on the X-ray radiation, set the integration time, and acquire an image of the object. Load the test image in MATLAB and display its histogram. Adjust the X-ray voltage and/or current and the integration time to get the correct exposure.
4. **Acquire the dark offset images for calibration:** While X-rays are turned off, acquire at least 10 dark offset images. Remember that the integration time should be the same as for all the other measurements (as determined in the above step).
5. **Start the image acquisition:** Turn on the X-rays and start the acquisition. This step will take the most time depending on the integration time (typically around 30 minutes). The system will rotate the sample about 360 degrees in 360 steps and after each step, it will take an X-ray image which is saved on the hard drive in raw format.
6. **Acquire the flat-field image:** Turn off the X-rays, remove the sample from the rotation stage, turn on the X-rays, and acquire the flat-field image. The input signal level should be similar to the one used in previous measurement but the detector must not be in saturation. Again, it is important to take more images because of the noise and statistical errors.
7. **Average the correction images:** Calculate the average of the dark offset and flat-field images to improve the quality.
8. **Normalize the images:** Perform pixel corrections on the acquired images.
9. **Make the sinogram.**
10. **Reconstruct the cross-section (slice).**

7 Images appendix



Figure 8: An aluminium profile on the rotation stage.

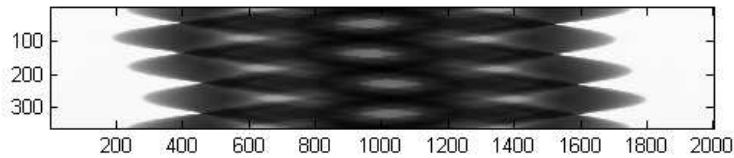


Figure 9: Sinogram of an aluminium profile.

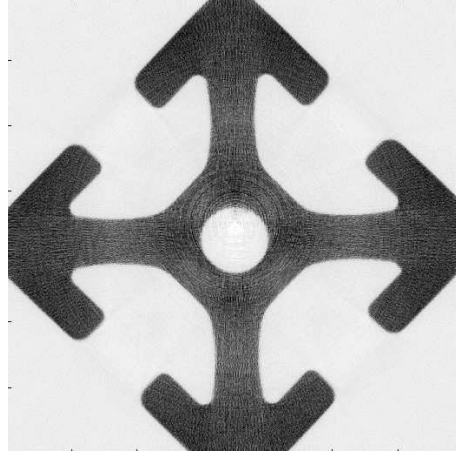


Figure 10: Final reconstruction of an aluminium profile.

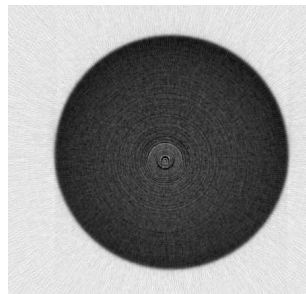


Figure 11: Final reconstruction of an aluminium cylinder.

8 Conclusion

During the course of the work, we managed to build a system which is now able to visualize the inner structure of bodies under investigation. We started from scratch having just the components of the resulting instrument – an X-ray detector and a source, a rotation stage, and a computer. The components were physically put together and software to control each of them was written. We used C language for controlling the detector and MATLAB for all other work. MATLAB was also the environment from which the system was controlled. We explored various effects on image quality of the reconstruction images. We found a couple of different artefacts which were present in the reconstructions and found a way how to limit them. Among others, let us mention the ring artefact and the influence of a correctly determined center of rotation.