Faster X-ray computed tomography in real-world dynamic applications

Graas, A.B.M.

Summary

Introduction and problem statement

This dissertation studies how the efficiency of Computed Tomography can be improved for industrial and scientific dynamic applications.

About Computed Tomography Computed Tomography (CT) is an imaging technique that makes it possible to look inside objects or the human body, using X-rays. This is done through the acquisition of X-ray images, called *radiographs*, taken from multiple angles around the object or body. For this purpose, a set-up is used that consists of a radiation source and X-ray detector, similar to an optical camera. A single radiograph, taken from one angle, shows how much radiation has passed through the materials, yielding a contrast, for example between bone (which absorbs a high quantity of X-rays) and tissue (which has a relatively low quantity of absorption). Since one radiograph does not contain sufficient information to form a 3D image, CT combines radiographs from often hundreds to thousands of angles. To achieve this, the radiation source and detector need to be rotated around the body, or an object can be positioned on a rotation stage. The number of angles determines, among other factors, the sharpness of the resulting 3D image.

The need for efficiency The computation time or computational power required to go from a set of radiographs to a 3D image—the so-called *CT reconstruction*—depends on the pixel resolutions of the radiographs and 3D image, as well as on the computational capabilities and the choice of algorithm. The filtered-backprojection (FBP) algorithm, for instance, is efficient enough to compute a reconstruction within just one second. The hardware implementation matters, too. Nowadays, graphics cards are widely used for CT, because they can perform computations on a part of the data synchronously with a single computer instruction. However, many modern X-ray applications require even faster algorithms and implementations for Computed Tomography, including:

- rapidly moving processes, in which each moment in time is effectively its own reconstruction problem;
- algorithms that require more computational power, for example because they must compensate for a low number of angles or a high noise level in the radiographs;
- applications that use CT continuously, for example for detecting production defects on conveyor belts, or for monitoring food safety;

- techniques based on artificial intelligence (AI), in particular neural networks, which must first learn from a large number of reconstructed examples before they can be applied to unseen data.

Summary of the chapters

Chapter 2 This chapter introduces an experimental setup consisting of three radiation sources and three detectors. Since this setup does not need to rotate, it is better suited for fast-moving processes, such as the fluidized beds used in the chemical industry. A fluidized bed consists of a mixture of gas and a particulate material, and behaves by approximation similar to a liquid. Existing techniques were previously only able to observe fluidized beds in a single cross-section at a certain height, or via a time averaged quantity. The advantage of the new technique is that it can track the morphological changes of gas bubbles inside fluidized beds both through time and the entirety of the 3D space.

Chapter 3 This chapter introduces a neural network that can handle the variable experimental data that is often encountered in *real-time* tomography. Real-time tomography makes it possible to observe an experiment live. Normally, neural networks need to learn from prior experiments before they are used. However, in the real-time tomography experiments that are performed at synchrotrons (particle accelerators), there is often insufficient time or suitable data prior to the experiment. The proposed method therefore learns during the experiment itself.

Chapter 4 This chapter introduces a software package called *ASTRA KernelKit*, which makes it easier to adapt tomographic algorithms for atypical reconstruction contexts. Existing tomographic algorithms are often only optimally efficient for standard problems, for example when the measurement data conform to commonly-used dimensions. The new software, based on the Python programming language, makes it easier for scientific users to create customized algorithms, experiment with them on graphics cards, and the software can use *kernel tuning* to find the fastest version for a predefined problem and hardware context.

Chapter 5 Finally, this chapter seeks a solution for removing noise from X-ray radiographs, which also improves the quality of subsequent CT reconstructions. State-of-the-art methods for noise removal are nowadays all data-driven, but generally require data sets of of noise-free examples from which to learn. For X-ray setups, such examples are not available in practice, because a noise-free radiographs require long measurement times. The presented solution therefore modifies the noise characteristics of radiographs acquired from modern X-ray scintillator detectors in such a way that each pixel subsequently becomes statistically independent. This statistical property then makes the data suitable for Blind-Spot Networks, a type of neural network for which noise-free examples are no longer needed.