Implementation of a performance optimized ray reassignment algorithm

Background:

With the available ultrafast X-ray electron beam computed tomography scanner of the HZDR up to 8000 non-superimposed cross-sectional images can be contactless obtained per second. Therefore, a free electron beam is circularly deflected onto a so called target, whereby a fast rotating X-ray source is generated. A radiation detector ring, arranged with a small axial gap, measures the radiation penetrated through the object at defined discrete X-ray point source positons p in an interval of $0 < \alpha \leq 2\pi$. In practice, the radiation detector system contains d = [1 … N_{Det}] single detectors being cyclically read-out with a constant sampling frequency f_{Samp}. Together with the constant electron beam deflection frequency f_{Defl} a certain number of radiation projections

$$N_{\text{Proj}} = \frac{f_{\text{Samp}}}{f_{\text{Defl}}}$$

are continuously acquired from the object. After the CT scan a multiple projection data set with $f_{\text{Defl}} \cdot t_{\text{Mess}}$ single CT data sets S_{dp}^{FAN} are obtained. These data are used to reconstruct the non-superimposed cross-sectional images using analytical, algebraic or stochastic reconstruction algorithms. However, for an effective implementation of standard reconstruction algorithms a ray reassignment must be performed. The single data sets, delivered by the CT system that uses a quasi-static point source and a detector arc geometry, are sorted as fan beam projections S_{dp}^{FAN} (see Figure 1b). From the mathematic view parallel beam projections $S_{a\theta}^{\text{PAR}}$ (see Figure 1a) are required.

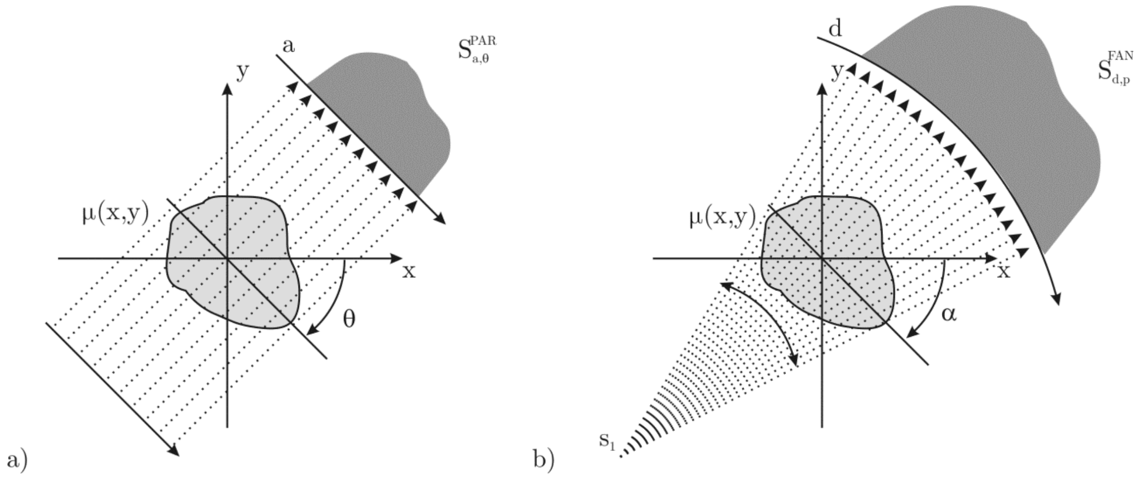

Figure 1: Acquisition of a) a parallel beam projection and b) a fan beam projection.

Task:

In a first step, the fan to parallel beam transfer matrix has to be calculated and stored as a so called hash table. Then, the projection data of *N* slices have to be resorted and interpolated into the parallel beam coordinate system. For an optimal usage of various resources, (e.g. number of CPUs and GPUs as well as correspondingly available host and device memory) the given specifications have to be considered in the programmed algorithm. Thus, the algorithm must decide:

- How many processes/streams have to be initiated?
- How many and which data must be transferred to and from the available device synchronously or asynchronously?

Contact:

André Bieberle (HZDR) | a.bieberle@hzdr.de | +49 351 260 2913