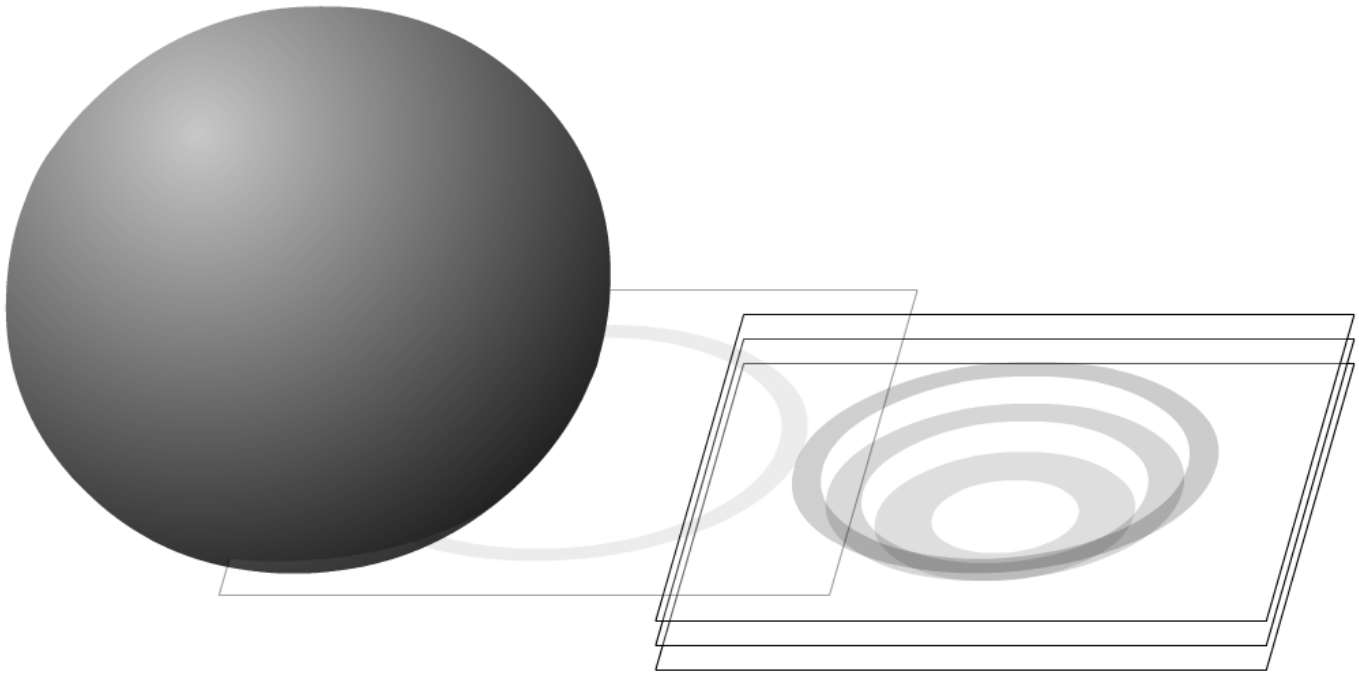
Parallel algorithm for 3D computed tomography problem

A.E. Kovtanyuk[1], E.I. Temnenko[2], and A.A. Khandorin[2]

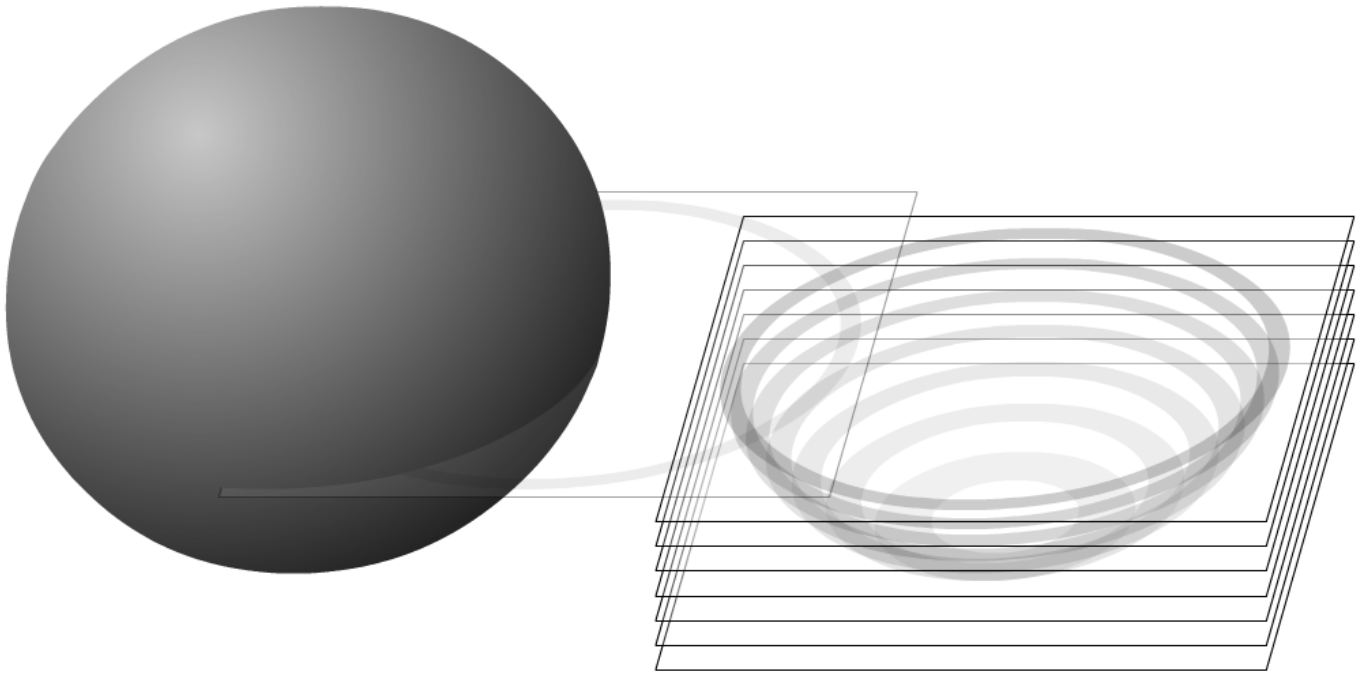
[1]Institute for Applied Mathematics, Vladivostok, Russia
[2]Far Eastern Federal University, Vladivostok, Russia

December 5, 2013

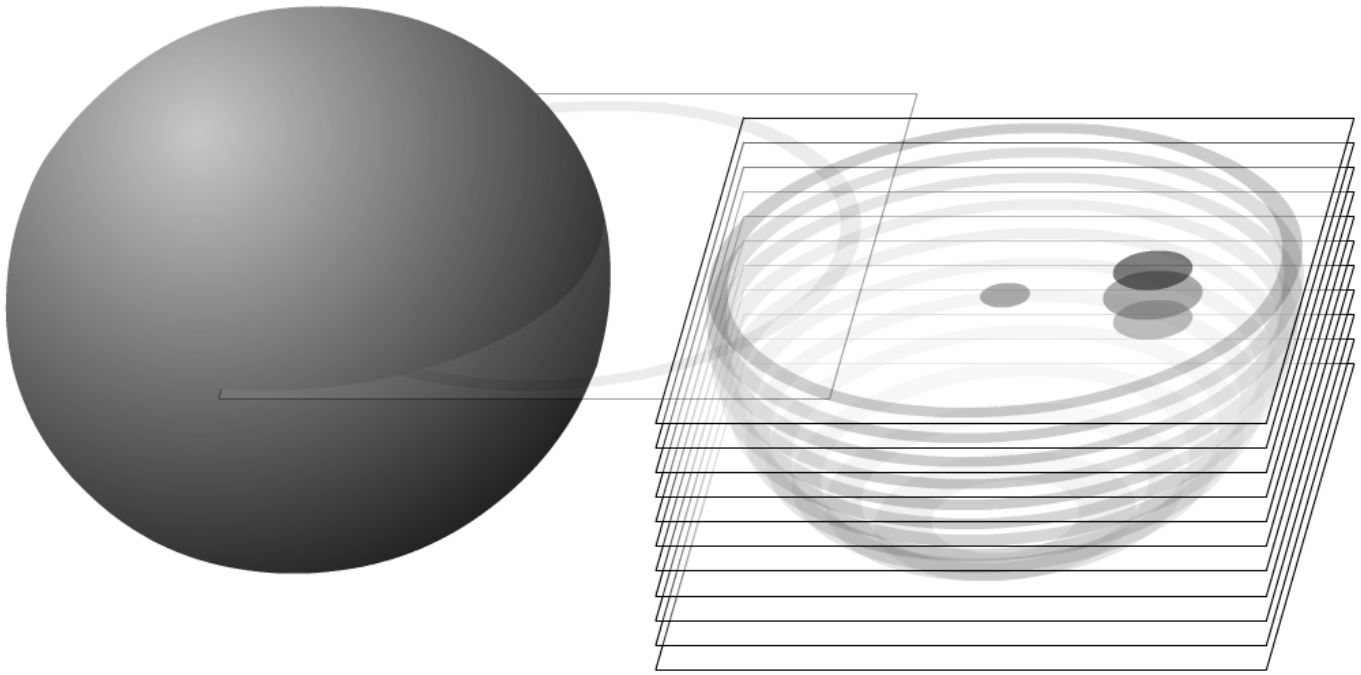
3D reconstruction:



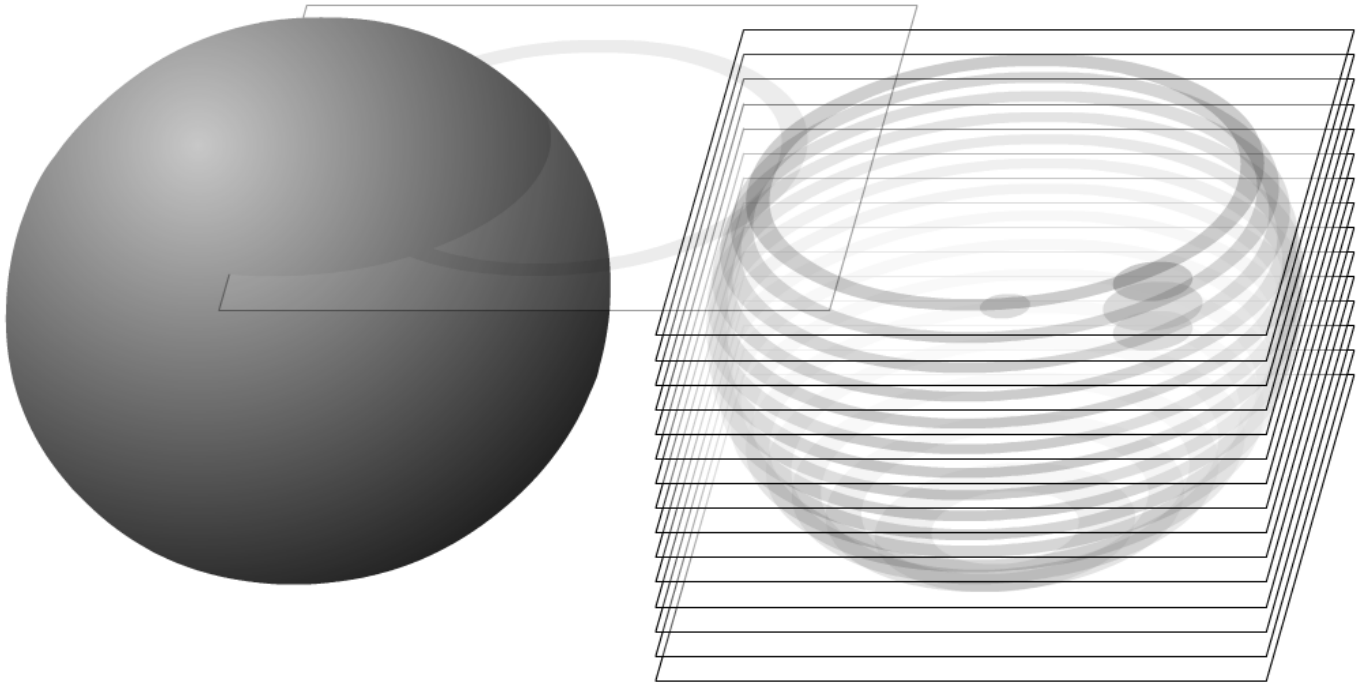
3D reconstruction:



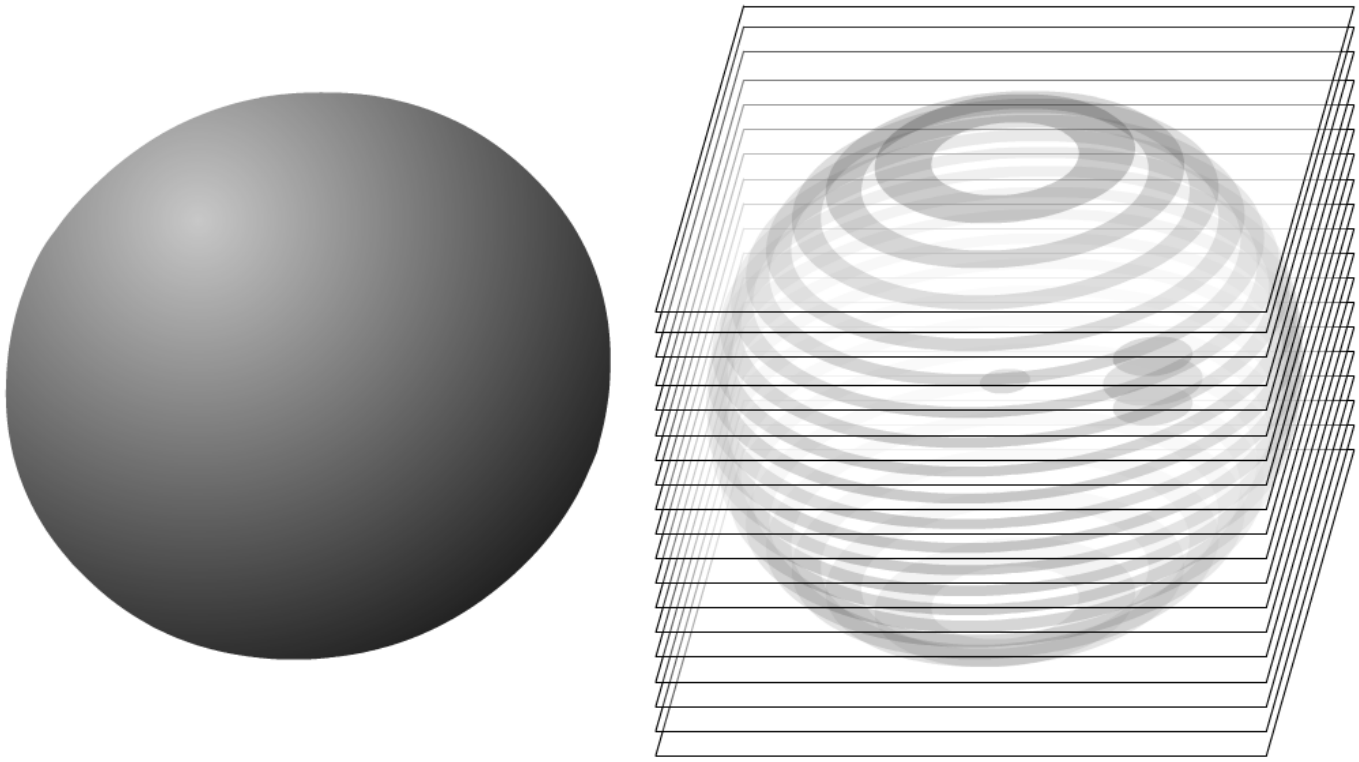
3D reconstruction:



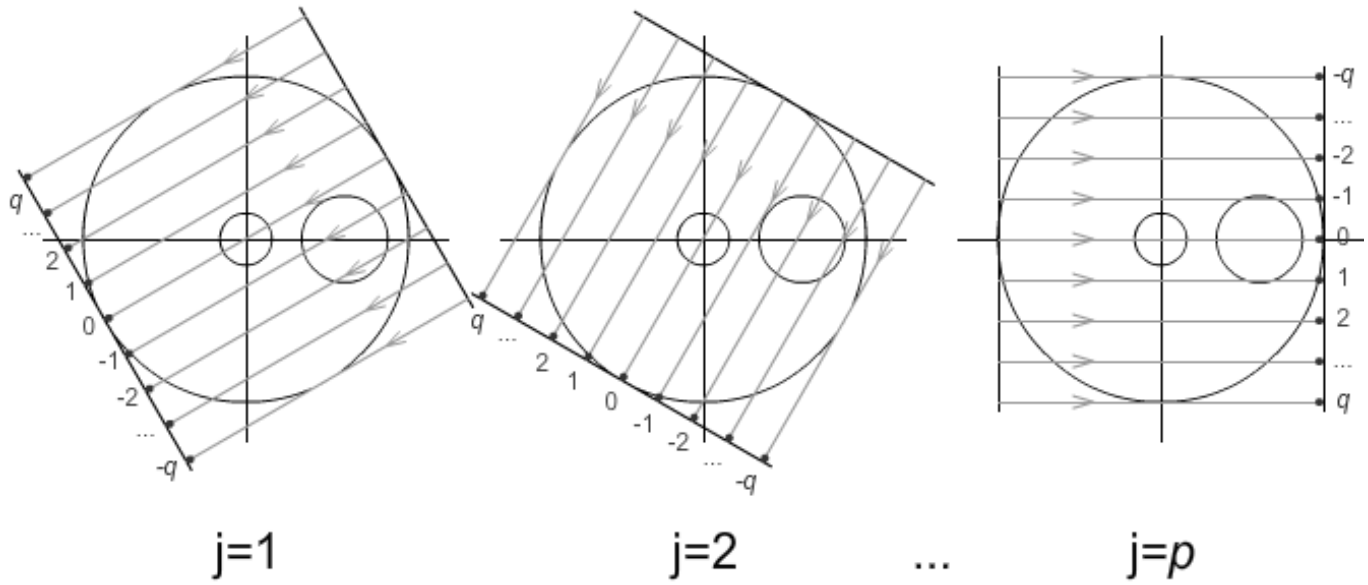
3D reconstruction:



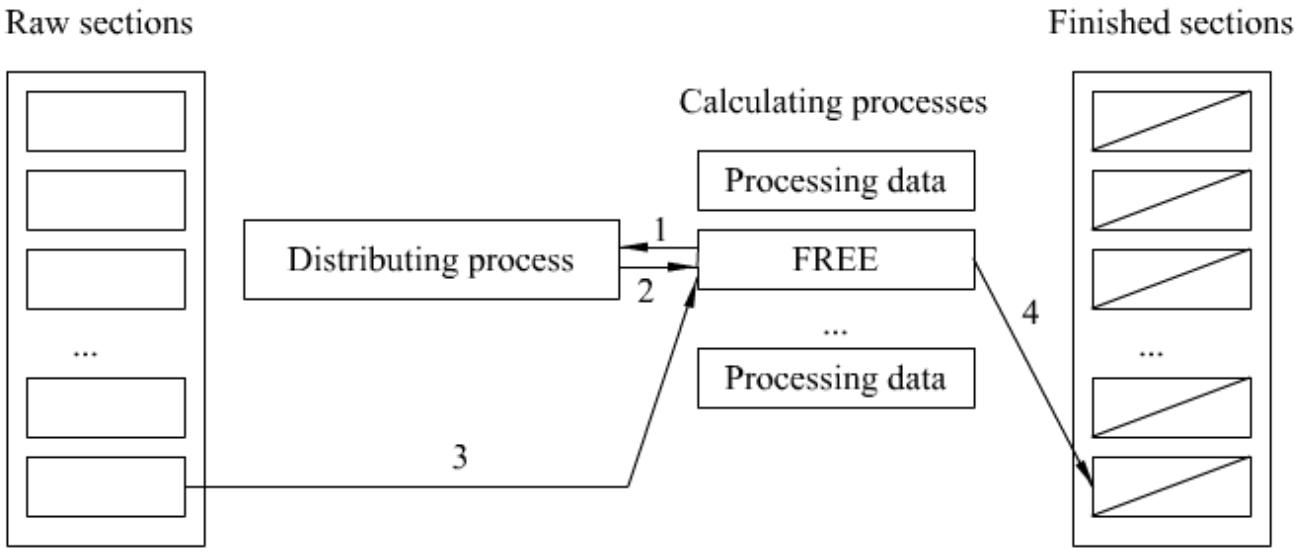
3D reconstruction:



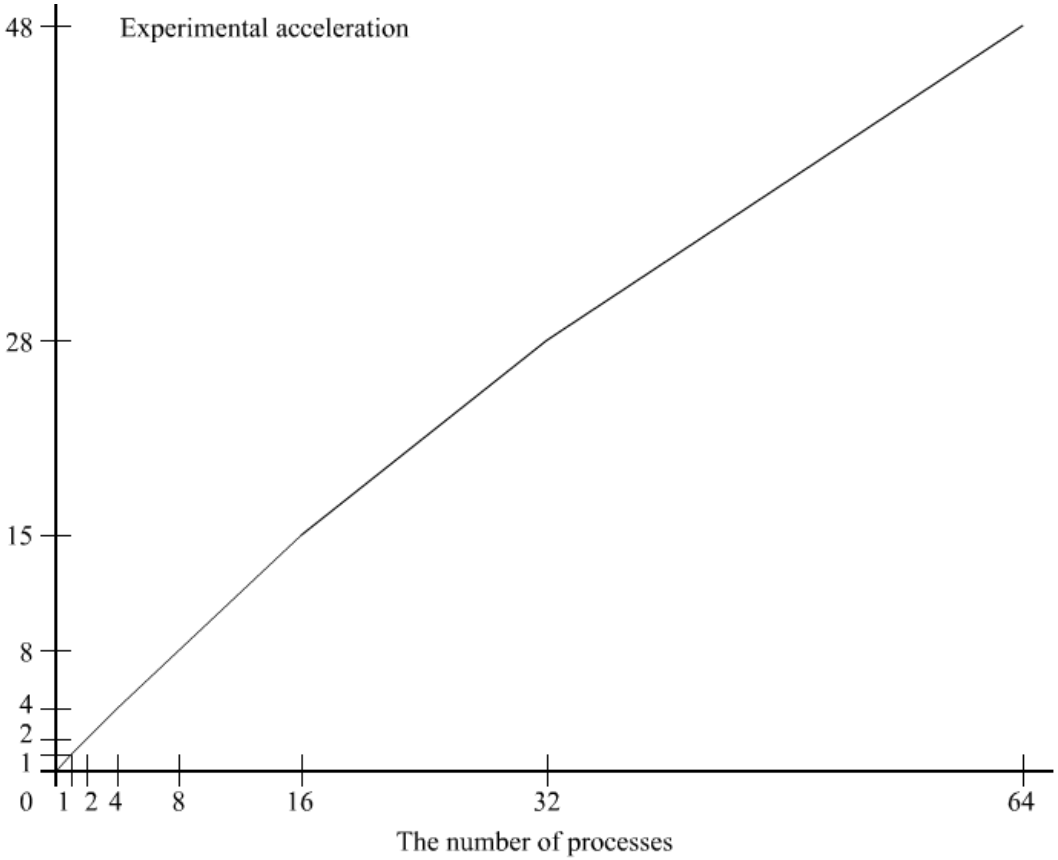
Irradiation at various angles: Parallel scheme the scanning



The scheme of the algorithm: Dynamic loading of compute nodes



The experimental acceleration



Efficiency of the algorithm

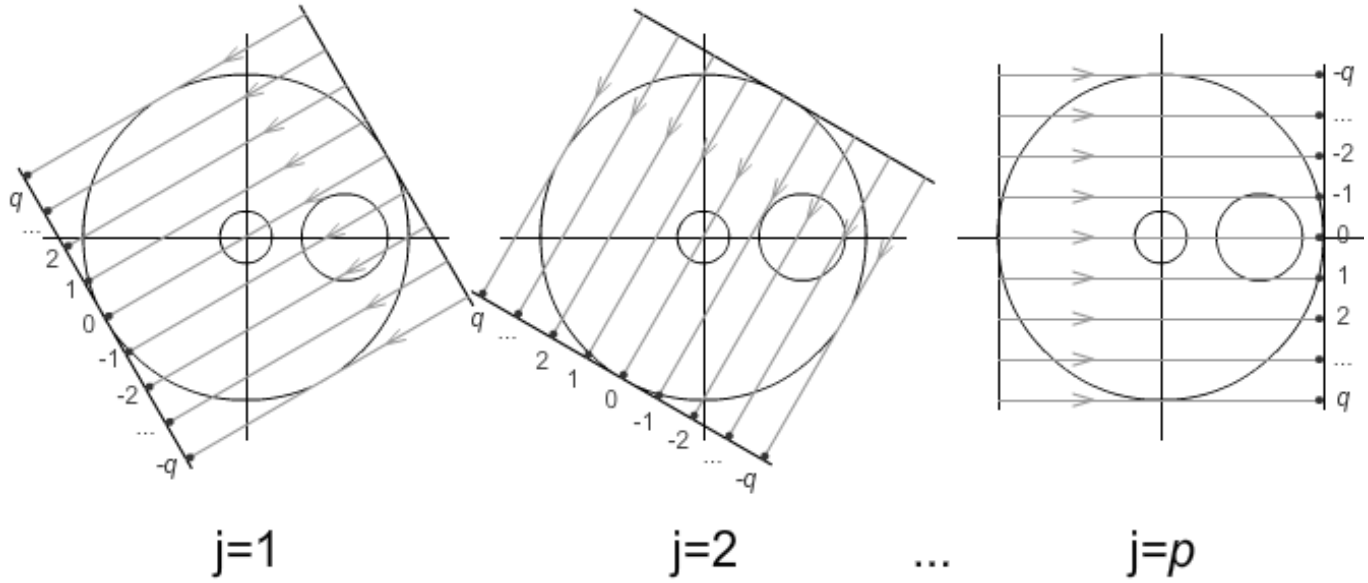
The number of processes	*Total time (sec.)*	*Experimental acceleration*	*The parallel fraction of the algorithm*
1	482,135	1	1
2	241,738	1,99445	0,9972
4	121,005	3,98442	0,9987
8	62,1094	7,76267	0,9956
16	31,4126	15,3486	0,9972
32	17,2215	27,9961	0,9954
64	9,94997	48,4559	0,9949

Implementation of the algorithm on GPU
(nVidia GeForce GTX 660M)

Algorithm of convolution and back projection

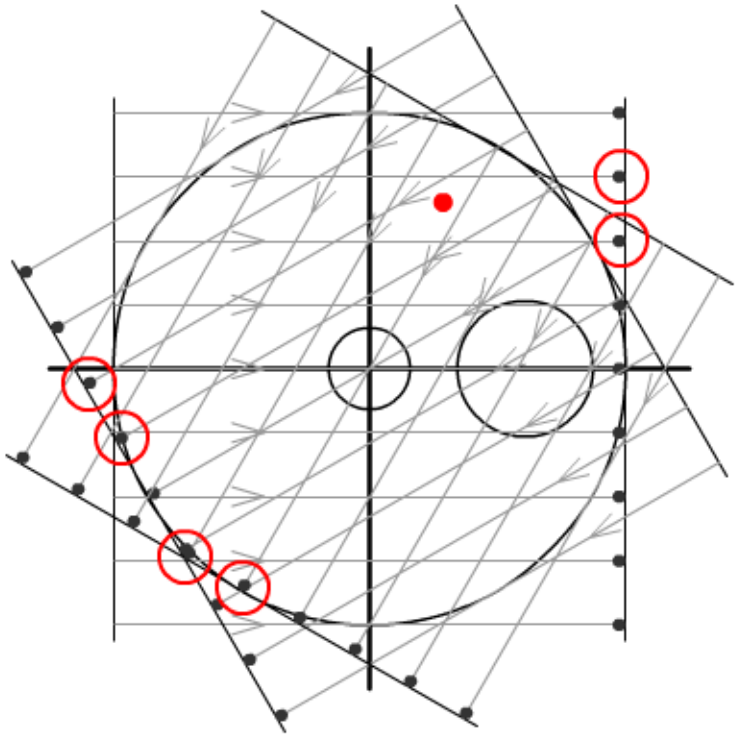
Step 1. Computing the convolutions:

$$v_{j,k} = h \sum_{l=-q}^{q} w_b(s_k - s_l) g(\theta_j, s_l), \quad j = 1, ..., p, \quad -q \le k \le q.$$



Step 2. Computing the back projection for $x = x(\theta_j, s)$:

$$f(x) = \frac{2\pi}{p} \sum_{j=1}^{p} ((1-u)v_{j,k} + uv_{j,k+1}), \quad k \le \frac{s}{h} < k+1, \quad u = \frac{s}{h} - k.$$



The run-time of the reconstruction

Cross-section number	Run-time (msec)
1	0.0685953
2	0.0661115
3	0.0545759
4	0.0370371
5	0.0359608
6	0.0317436
7	0.0291794
8	0.0292195