

Improvement of X-Ray image acquisition using a GPU based 3DCT simulation tool

M.Reiter ^a, M.M.Malik ^b, C.Heinzl ^a, D.Salaberger ^a,
E.Gröller ^b, H.Lettenbauer ^c, J.Kastner ^a

^a Upper Austrian University of Applied Sciences, Wels Campus, AUSTRIA

^b Vienna University of Technology, Institute of Computer Graphics and Algorithms, AUSTRIA

^c Carl Zeiss Industrielle Messtechnik GmbH, Oberkochen, GERMANY

Abstract

This paper presents a simulation tool for industrial X-Ray computed tomography (CT) systems which is able to predict the results of real measurements. Such a prediction helps the technician in measurement technology to minimize artefacts by using optimal measurement parameters and therefore it helps to get more accurate results. The presented simulation software offers an implementation for CPU's and GPU's. The performance difference between these implementations is shown, for a specific test part. Furthermore a parameter variation has been carried out, to illustrate the influence of the acquisition settings. We use a multi-image view tool to compare and evaluate the acquired dataset series which contains CT data gained with different X-Ray source voltages and a different number of projections.

Keywords: Industrial X-Ray Imaging, Computed tomography, CT, Simulation

1. Introduction

X-Ray Imaging is a common method for non-destructive testing using radiographic or tomographic systems. A new and challenging application in the field of computed-tomography (CT) is metrology, which has to fulfil the demands of today's standards for quality control. The achievable uncertainty of measurement mainly depends on the CT data's quality and in addition on the used evaluation algorithms. CT device settings like X-Ray source voltage, anode current, filter plates, detector integration time and the measurement conditions influence the image acquisition and the quality of the CT dataset. Properly chosen settings produce sharp edges, minimise artefacts, and simplify subsequent semi automatic evaluation procedures within CT data. Selection and optimisation of these parameters has to be done by an experienced system operator, which leads to subjective measurement results. A CT simulation software would be able to do this optimization process automatically and generate user independent device parameters and results. Monte-Carlo simulations are able to model the interactions of X-Rays with matter very accurately, but due to their high computational costs, Monte-Carlo simulations are not suitable for the required optimisation of a complete CT scan with a high

number of projections. The objective of the presented work is a time optimized simulation for industrial cone beam CT's, which models all essential interactions at a sufficient accuracy and is applicable for the optimization of the device parameters.

2. Polychromatic X-Ray cone beam computed-tomography (3DCT)

An industrial cone beam CT system as shown in Figure 1 consists of an X-Ray source, a rotary table and an X-Ray sensitive detector. During a discrete 360 degree turn of the specimen, penetration images which contain the spatial X-ray attenuation are recorded. Using these images it is possible to reconstruct the fully inner and outer structure of the specimen. The reconstruction method used within this work is a filtered back-projection.

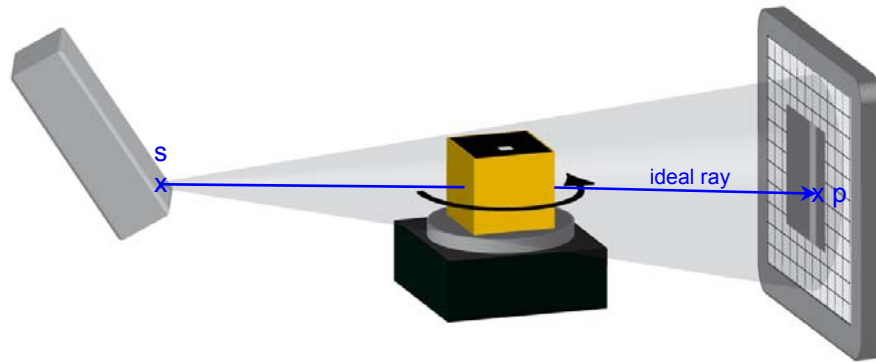


Figure 1: Schematic of a cone beam computed-tomography system. s is the source and p the end point of an ideal casted X-ray penetrating the object.

CT measurements for this investigation were performed by a RayScan 250 E CT device consisting of a Viscom 225 kV-microfocus tube and a 1024x1024 Perkin Elmer flat panel detector.

3. Modelling a polychromatic X-Ray cone beam computed-tomography system

Modelling of a cone beam CT system include, the X-Ray source model, the interactions of X-Rays with matter and the detector model. The presented simulation uses an X-Ray source model published by Ebel et al. [1]. It allows the calculation of polychromatic spectra consisting of Bremsstrahlung and characteristic radiation. Figure 2a illustrates a calculated spectrum. The virtual specimen is represented by a surface model consisting of a triangle mesh and an assigned material, which is defined by mass weighted chemical formulas. Furthermore the simulation is able to handle more than one surface mesh and even different material assignments.

A Raytracing algorithm [2] calculates the penetration lengths $L_{(s,p,N)}$ along an ideal ray from the source s to a point p on the detector, whereas the whole sensor area has to be covered with ideal rays. Subsequently Lambert-Beer's law according to Equation (1) is applied for each ray and each energy channel ΔE . Finally the photons in a specific energy channel are weighted with $DE_{(E)}$ the detector efficiency (Figure 2b), and summed

up to a grey value GV (Equation (2)). $P_{0(E)}$ and $P_{(E)}$ are the number of X-Ray photons before and after the linear attenuation process, μ_N are linear X-Ray attenuation coefficients as given by Brunneti et al.[3] or combinations of them. Effects caused by beam hardening are completely covered.

$$P_{(s,p,E)} = P_{0(s,p,E)} \cdot e^{(-\mu_{1(E)} \cdot L_{(s,p,1)} - \mu_{2(E)} \cdot L_{(s,p,2)} \dots - \mu_{N(E)} \cdot L_{(s,p,N)})} \quad (1)$$

$$GV_{(s,p)} = \sum_{E=E_1}^{E_{max}} P_{(s,p,E)} \cdot DE_{(E)} \quad (2)$$

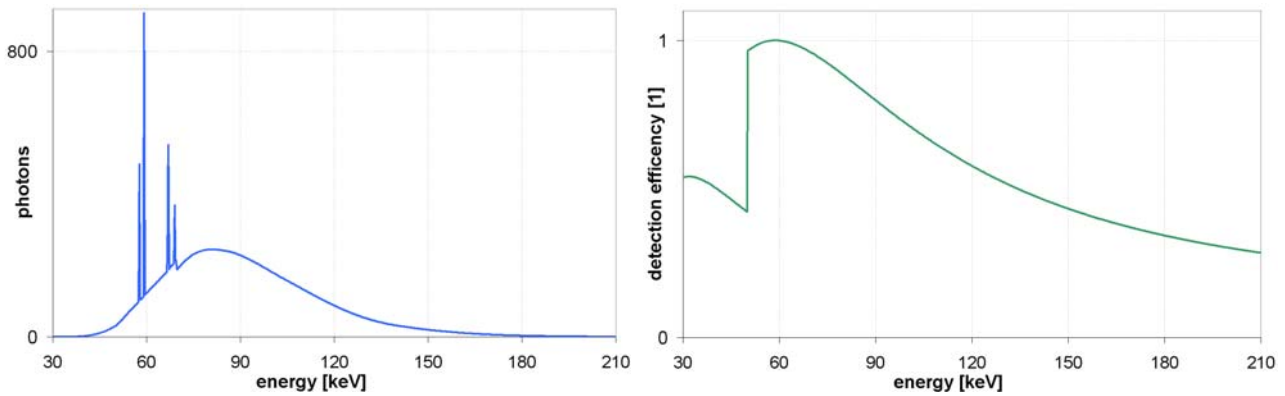


Figure 2: (a) Calculated X-ray spectrum: $U=210\text{keV}$, $I=300\mu\text{A}$, $\Delta T=1\text{s}$, $\text{Cu } 1\text{mm}$, $\Delta E=1\text{keV}$, tungsten target (b) detector efficiency model used for an Perkin Elmer RID1640 Al1 ES X-Ray detector which has a scintillating screen made of $\text{Gd}_2\text{O}_2\text{S:Tb}$.

Due to the high number of detector pixels ($>1024 \times 1024$), pixel oversampling and the high number of energy channels (common value for $\Delta E = 1\text{keV}$), Equation (1 and 2) have to be calculated for several million times to generate suitable results for one projection image. The calculation for each ray is independent, therefore it can be parallelised on a multi-processing unit.

In addition to Lambert-Beer's law the simulation tool offers the opportunity to consider first order scatter effects within the inspected specimen. With some approximations it is possible to reduce the scattering process to N_E convolutions, where the operator-kernel is about twice the size of the detector dimension and N_E is the number of energy channels for the scatter calculations (common value for $\Delta E = 20\text{keV}$). In contradiction to conventional analytic first order scatter models (e.g. Freud et al. [5]) our approach does not require a representation of the specimen in voxels. Monte-Carlo Simulations [6][7] and hybrid approaches [8][9] are out of scope, because of their high calculation effort.

Furthermore a model for detector inherent scattering has been developed, in which the time-consuming tasks are N_E convolutions with point spread function (PSF) kernels up to about 300×300 pixel for a Perkin Elmer RID1640 Al1 ES flat panel detector. Considering one specific detector, the kernel size depends only on the participating X-Ray photons energy. For each energy channel N_E a kernel has to be specified.

4. Implementation and Performance

The whole CT Simulation process has been implemented on the CPU using C++ and on the GPU using Nvidia CUDA toolkit [10]. Algorithm 1 shows pseudo code for the Lambert-Beer's law implementation using a CUDA Kernel within C++. The code demonstrates the necessary steps to use the graphics card for general purpose calculations. Starting with the device memory allocation and continuing with the transfers between device and host. The Memory management on the device is very critical due to the limited memory available on graphics boards. Constant memory is a read-only memory and accessible by all threads, shared memory is useable within a thread and global memory is read and write able by all threads, but the access times are

Algorithm 1: Pseudo Code of Lambert-Beer's Law using the CUDA Toolkit within C++. Using the terminology of CUDA, a device is the graphics card and the host is the system using the graphics card.

```
void ImGen_kernel_wrapperForCpp( ... )
{
    Allocate Memory on the DEVICE;
    Transfer data from HOST to DEVICE;
    Define textures for read-only data in the global memory of the DEVICE;

    Split the rays toward the detector into threads per block dimBlock and
    blocks per grid dimGrid;
    sharedMemory used per thread has the size of the xray source spectra;

    ImGen_kernel<<<dimGrid,dimBlock,sharedMem>>>(Image);

    Transfer Image from DEVICE to HOST;
    Free Memory on the DEVICE;
}

__global__ void ImGen_kernel( Image )
{
    Determine the current ideal ray via block and thread id;
    Load initial xray source spectrum  $iniS_{(E)}$  from constant to shared memory
     $curS_{(E)}$ ;

    Iteration through absorbing objects
         $curS_{(E)} = \text{LinearAttenuation}(curS_{(E)}, \mu_{(E)}, length);$ 

    Image(ray) = Sum of photons within the spectrum  $curS_{(E)}$ ;
}

__device__ void LinearAttenuation( spectrum(E),  $\mu(E)$ , length )
{
    Iteration through the energy channels E
    spectrum(E) = spectrum(E) *  $\text{__expf}(-length * spectrum(E) * \mu(E))$ ;
}
```

potentially 150 times longer than to registers, constant and shared memory. NVidia graphics cards supporting Compute Capability 1.3 have 16kB of shared memory, 64kB of constant memory [13], which limits the spectrum size to 4096 32Bit float values. Calling a CUDA Kernel require a division of the calculations into parallel calculateable parts so called threads. That's why every thread calling the kernel function first has to determine the part it has to calculate via the block and thread-id. The number of threads is limited [13]. Therefore blocks of threads are specified. After running the kernel the calculated results have to be transferred to the host and the memory on the device can be freed.

We performed a performance comparison of the CPU and GPU implementation of Lambert-Beer's law for a specific test part called TK03. This aluminium test part TK03 shown in Figure 3, consists of one central drill hole, 6 small inner drillholes of which four are located in the bottom section of the specimen and two at the top section. Figure 3b shows the projection image of TK03 used for the performance comparison. Further Figure 2a and 2b show the used spectrum and detection efficiency.

The calculation times for one projection are shown in Table 1. They depict a performance increase by a factor of 7 to 9, when using the GPU instead of the CPU implementation. This increase in performance shows a slight dependence on the number of ideal rays. The benchmark was done on an Intel Quad core QX9650 system using an NVidia G80 GPU.

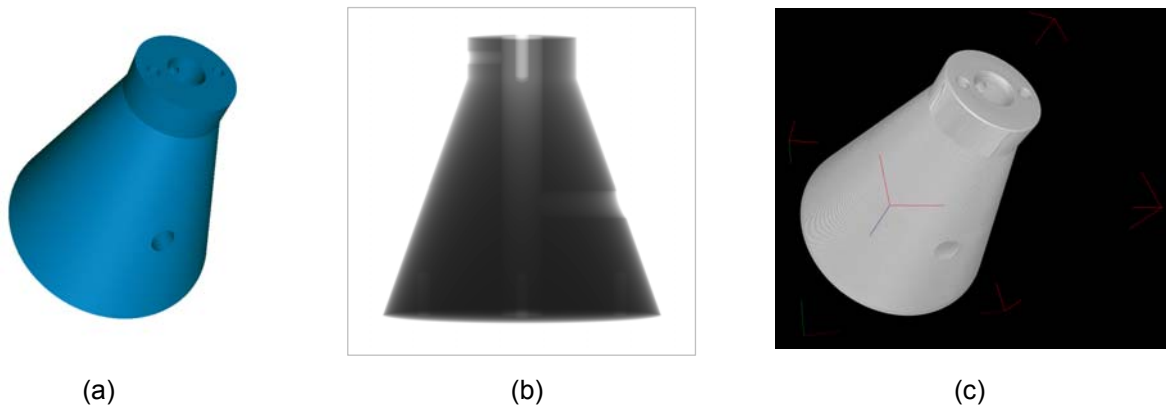


Figure 3: (a) Surface model of the aluminium test part TK03 for the evaluation of the simulation tool consisting of 200000 triangles, (b) Simulated X-Ray projection images used for the reconstructed image, (c) 3D-representation of the simulated CT data of the test part TK03.

Table 1: Performance comparison solving Lambert-Beer's Law using a CPU or a GPU implementation for a different number of rays per projection.

no. of ray	C++ OpenMP CPU Intel QX9650	CUDA GPU NVidia G80	performance difference
512x512	608ms	80ms	7.6x
1024x1024	2420ms	350ms	6.9x
2048x2048	9,71s	1,1s	8.8x
4096x4096	39,04s	4,46s	8.7x

The time critical steps for the scatter models mentioned in chapter 3 are large inseparable convolutions in the Fourier space. We use a FFT-convolution published within the CUDA SDK by V.Podlozhnyuk [10]. Compared to other available CPU implementations (e.g. Intel Math Kernel Library MKL, fftw3) the CUDA implementation shows a 4x shorter computation time for the listed hardware (Table 2).

Table 2: Performance comparison of single-precision, complex, out-of-place, 2d FFT implementations.

implementation	machine	FFT size	computation time
Intel MKL**	CPU Intel P4 3GHz	1024x1024	46ms
NVidia CUDA***	GPU NVidia G80	1024x1024	10ms*
fftw3**	CPU Intel P4 3GHz	1024x1024	54ms

* including transfers between host and device, ** [12], *** [10]

The implementation of Lambert-Beer's law within C++ using OpenMP and all further implementations, which are necessary for a CT-Simulation are not discussed within this paper.

5. Results

Figure 4 shows simulated and measured CT data of test part TK03. This comparison of real and simulated CT reconstructions shows a slight difference regarding the strength of cupping artefacts toward the centre of the cylinder.

To demonstrate the potential of this simulation tool regarding an optimization of the imaging parameters a parameter variation has been carried out for the voltage of the X-Ray source and the number of projections. Table 3 shows some details about the acquired datasets. Differences within these datasets are compared and visualised using a multi-image view implemented by M.M.Malik et al.[4], which overcomes a side by side comparison of the dataset series. Figure 5a and 5b show a multi-image view of test part TK03, which divides a slice image into hexagons. In the centre of every hexagon the so called base-dataset is displayed. The surrounding sections contain the datasets that are compared to the base-dataset. Figure 5c visualises the grey value difference between base-dataset and the dataset within a specific section. This difference is colour coded according to a user defined colour scale. The grey value of air is almost constant in all datasets in the region highlighted by a green rectangle (Figure 5b) for a variation of the X-Ray source voltage in the steps 150kV, 180kV, 210kV and 220kV. However Beam Hardening has raised the air grey values within

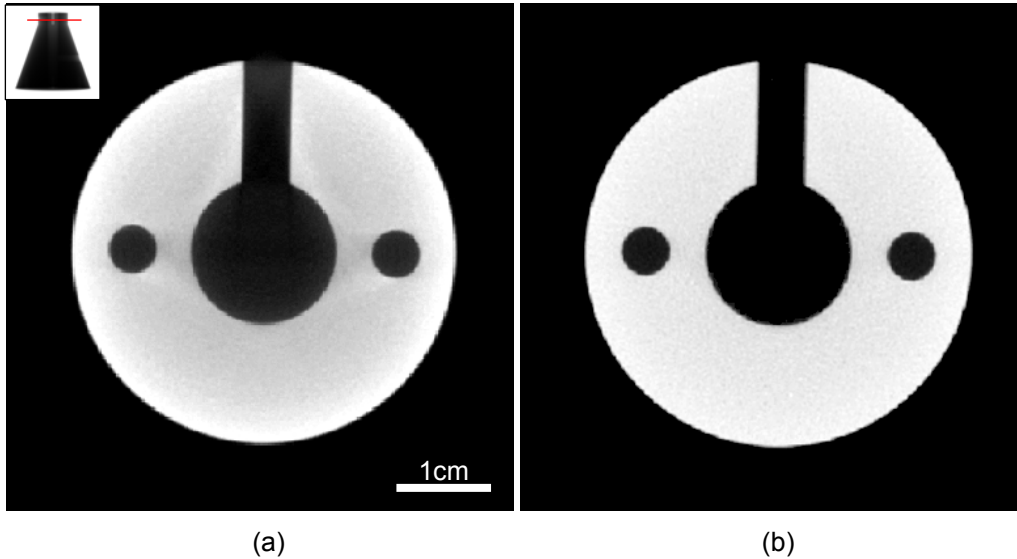


Figure 4: Axial slice within reconstructed data of real (a) and simulated (b) CT acquisitions of TK03.

the yellow region causing only a small difference between aluminium and air, where the penetration lengths are highest during the acquisition. Figure 5b and 5c show that beam hardening is less using a higher X-Ray source voltage, which leads to the conclusion that 150kV is a too low X-Ray source voltage for the inspection of test part TK03.

Reconstruction results with a variable number of projections are shown in Figure 6. The reconstruction has been done with 90, 270, 360, 720, 990 and 1440 projections. Figure 6b shows an increasing quality until 720 projections, further projections cause a negligible quality improvement. Acquiring 720 projections instead of 1440 will reduce the real measurement time to 50 percent, without a significant loss in image quality.

Table 3: Dataset series of test part TK03. The first dataset series contains a variation of the X-Ray source voltage. The second series show the influence of the used number of projection for a CT acquisition.

no.	parameter	dataset size	voxel size	no. of datasets
1	X-Ray source voltage	16bit uint 1000x1000x882 (1,64GB)	122 μm	4
2	no. of projections	16bit uint 1000x1000x882 (1,64GB)	122 μm	6

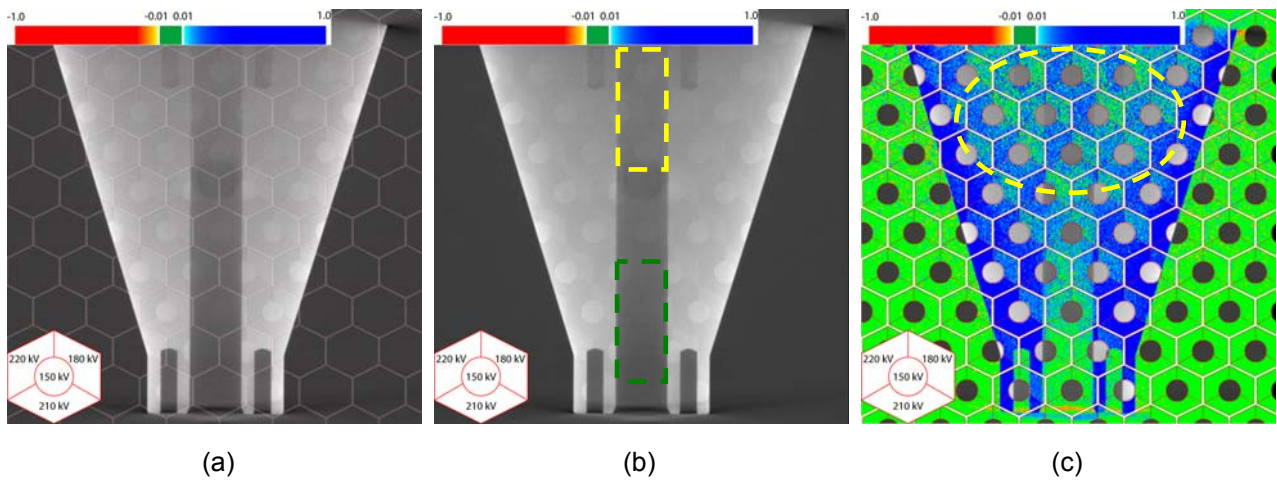


Figure 5: (a) Sagittal multi-image view of parameter variation no.1: X-Ray source voltage, (b) without the borders of the hexagons, (c) colour coded difference between surrounding datasets and base-dataset

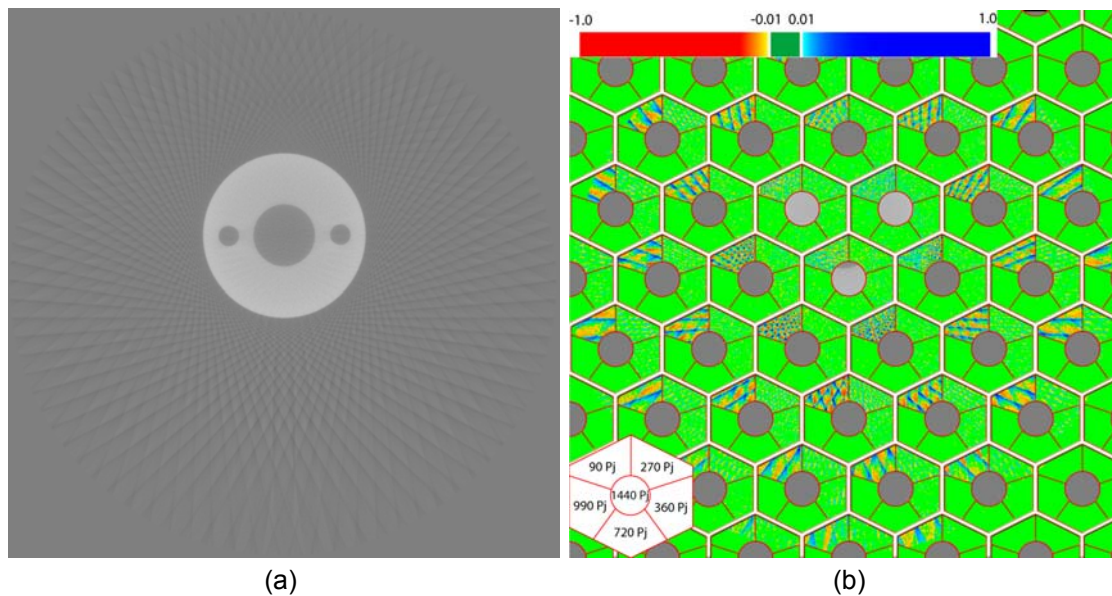


Figure 6: Multi-image view of dataset series no.2. (a) Axial slice view of the dataset with 270 projections, (b) colour coded difference between different simulated CT datasets generated by various numbers of projections.

6. Conclusion

We have presented a simulation tool for industrial X-Ray Computed Tomography systems, where we take advantage of the graphics hardware to achieve minimal calculation times. We have shown CT results caused by different acquisition parameters, using a multi-image view and we were able to choose optimal measurement parameters.

Acknowledgements

This work was supported by the SimCT project of the Austrian Research Promotion Agency, grant number 812136-SCK/KUG.

References

- [1] H.Ebel, XRS, 1999, 28,4, 255-266, and H.Ebel , „Advances in X-ray Analysis“, 2006, 49, 267-273.
- [2] T.Möller, B.Trumbore, „Fast, Minimum Storage Ray-Triangle Intersection“, Journal of Graphics Tools, vol. 2, 21-28, 1997
- [3] A.Brunetti, M. Sanchez del Rio, B. Golosio, A. Simionovici, A. Somogyi, Spectrochim Acta, 2004, B 59, 1725– 1731.
- [4] M.M.Malik, E.Gröller, Visual Exploration for Parameter Studies of Industrial CT Dataset Series, Vienna University of Technology, Institute of Computer Graphics and Algorithms, technical report 12/2008
- [5] N.Freud, P.Duvauchelle, S.A.Pistrui-Maximean, J.-M. Létang, D.Babot, Deterministic simulation of first-order scattering in virtual X-ray imaging, Nuclear Instruments and Methods in Physics Research B 222 (2004) 285–300
- [6] M.Mantler, B.Chyba, M.Reiter, Monte Carlo Simulation of Projections in Computed Tomography, Denver X-ray Conference 2007, Denver (US) 30.7-3.8 2007
- [7] G.-R.Jaenisch, C.Bellon, U.Samadurau, M.Zhukovskiy, S.Podoliako, McRay - A Monte Carlo Model Coupled to CAD for Radiation Techniques, European Conference for NDT 2006, Berlin, Tu.4.3.3
- [8] N. Freud, J.-M.Le´tang, D.Babot, A hybrid approach to simulate multiple photon scattering in X-ray imaging, Nuclear Instruments and Methods in Physics Research B 227 (2005) 551–558
- [9] J.Tabary, R.Guillemaud, F.Mathy, A.Glière, P.Hugonnard, Combination of high resolution analytically computed uncollided flux images with low resolution Monte Carlo computed scattered flux images, Proc. IEEE-MIC, Norfolk, November 12-16 2002.
- [10] www.nvidia.com/cuda
- [11] V.Podlozhnyuk, FFT-based 2D convolution, CUDA SDK Manual, 2007
- [12] <http://www.fftw.org/speed/Pentium4-3.06GHz/>
- [13] NVidia CUDA Programming Guide 2.0, 6/7/2008