doi:10.1017/S1431927618014551
Microsc. Microanal. 24 (Suppl 2), 2018

A Fast, Flexible and Accurate Monte-Carlo Tool for Simulating X-Ray Fluorescence Tomography

Kian Shaker[1*], Jakob C. Larsson[1], and Hans M. Hertz[1]

[1.] Department of Applied Physics, KTH Royal Institute of Technology/AlbaNova, 106 91 Stockholm, Sweden
* Corresponding author, kian.shaker@biox.kth.se

X-ray fluorescence (XRF) tomography is an upcoming and promising imaging modality. By combining x-ray imaging (CT) with targeted nanoparticles (NPs) it has the potential for high-resolution molecular medical imaging. The major challenges today include long scan times and high radiation doses due to the limited effective flux available with conventional laboratory sources, forcing most implementations of today to rely on synchrotrons.

A laboratory XRF tomography setup suitable for small-animal imaging has been developed over the last couple of years by the BIOX research group at KTH, Stockholm [1]. Recent results include 3D 100-µm-range resolution localization of tumors in mice, using passively targeted molybdenum NPs [2]. In order to optimize the experimental parameters and to investigate the limitations of the system, Monte-Carlo (MC) simulation tools are highly valuable.

Today there are many general MC tools available for simulating x-ray imaging, such as PENELOPE [3], Geant4 [4] etc. However, due to the complex physics, most general codes are inherently slow and implement little to none parallelization. GPU-parallelization has provided massive improvements in computational speed, however, it is almost always combined with a physics model of reduced complexity, hence trading generality for speed. We developed a GPU-based XRF code ("XRF-GPU") which is to our knowledge the only MC tool developed for providing fast, flexible and accurate simulations of XRF tomography.

XRF-GPU is a heavily modified version of MC-GPU [5], originally a code for simulating cone beam CT. As the original code disregards the emission of x-ray fluorescence, it has been implemented in XRF-GPU. Furthermore, the present code allows for many customizations regarding detectors; geometry, energy resolution, thickness and material of active detector material etc. Additionally, the incident x-ray beam shape and spectrum is fully customizable. The code can simulate any voxelized object and can handle many different materials. For a tomography, the user can specify the acquisition geometry; number of rotations as well as horizontal and vertical translations. Data is saved as spectra at the detectors for each step in the tomography. Registered photon counts produced from fluorescence are stored separately from the rest, making it possible to separate the fluorescence signal from the background if desired. No post-processing or filtering of the simulated output is necessary. The simulations are directly comparable to quantitative experimental data.

In order to validate results from XRF-GPU with experiments, measurements on a phantom was carried out. Figure 1a displays a top view of the experimental arrangement, consisting of two detectors measuring absorption and fluorescence respectively. A narrow pencil beam is used to scan the object and excite regions with molybdenum content, which consequently emit characteristic fluorescence. The object is a cylinder made out of PET with holes filled with water and different concentrations of molybdenum (see

Fig. 1b). For a more thorough explanation of the experimental arrangement, see [1,2]. A tomography was then performed with 360 projections, 200 µm translation steps with 200 ms exposure time. The same object and scenario was then simulated using XRF-GPU on a PC with a GTX-1050Ti NVIDIA GPU.

The tomographic datasets were reconstructed using in-house iterative algorithms based on a simultaneous iterative approach (SIRT). Furthermore, the algorithms take into account effects of self-absorption, meaning that we can reconstruct quantitative maps of elements that emit detectable fluorescence. The resulting molybdenum maps are shown in Fig. 2. Remarkable agreement is displayed both qualitatively and quantitatively between simulated and experimental tomographies. The overall simulated flux in XRF-GPU was $4 \cdot 10^9$ ph/min (the whole tomography taking 47 min to simulate). For comparison, the number of photons simulated per time is $> 10^3$ times faster than in PENELOPE.

In conclusion, we have developed XRF-GPU; a MC tool for simulating XRF tomography. Simulations have been validated to good accuracy against phantom experiments. Compared to PENELOPE, a well-known reference MC code capable of simulating the same scenario, it is $> 10^3$ faster, allowing XRF tomographic simulations in reasonable-time frames [6].

References:

[1] H. M Hertz *et al*, Opt. Lett., **39** (2014), p. 2790-2793
[2] J. C. Larsson *et al*, submitted.
[3] F. Salvat *et al*, OECD/NEA (2014)
[4] S. Agnostinelli *et al*, Nucl. Instrum. Methods Phys, **506** (2003), p. 250-303
[5] A. Badal and A. Badano, Medical Physics, **36** (2009) p. 4878-4880
[6] The authors acknowledge funding from the Wallenberg Foundation.

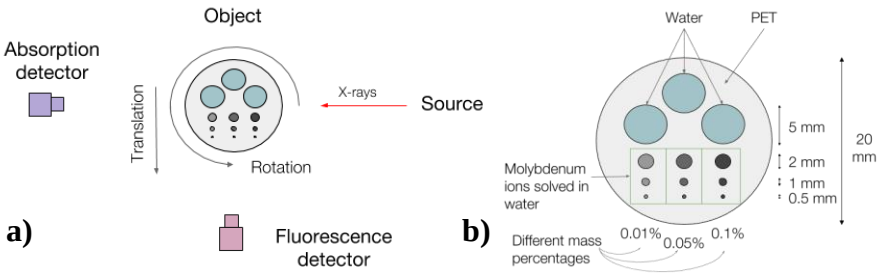

Figure. 1. **a)** Top view of the experimental arrangement (see [1,2] for more details). **b)** Top view of cylindrical phantom with holes used for experimental validation. Solutions with different mass percentages of molybdenum were used to fill the three bottom rows.

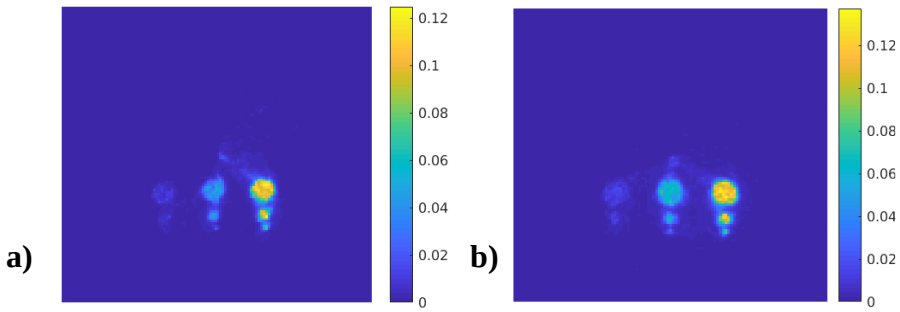

Figure. 2. Reconstructed molybdenum maps from tomographic datasets acquired from **a)** experiment and **b)** XRF-GPU. Colorbar displaying reconstructed mass percentages of molybdenum, as shown in Fig. 1b.

https://doi.org/10.1017/S1431927618014551 Published online by Cambridge University Press