

Please cite the Published Version

Golosio, B, Schoonjans, T, Brunetti, A, Oliva, P and Masala, GL (2014) Monte Carlo simulation of X-ray imaging and spectroscopy experiments using quadric geometry and variance reduction techniques. *Computer Physics Communications*, 185 (3). pp. 1044-1052. ISSN 0010-4655

DOI: <https://doi.org/10.1016/j.cpc.2013.10.034>

Publisher: Elsevier

Version: Accepted Version

Downloaded from: <https://e-space.mmu.ac.uk/623354/>

Usage rights:  [Creative Commons: Attribution-Noncommercial-No Derivative Works 4.0](https://creativecommons.org/licenses/by-nc-nd/4.0/)

Additional Information: This is an Author Accepted Manuscript of a paper accepted for publication in *Computer Physics Communications*, published by and copyright Elsevier.

Enquiries:

If you have questions about this document, contact openresearch@mmu.ac.uk. Please include the URL of the record in e-space. If you believe that your, or a third party's rights have been compromised through this document please see our Take Down policy (available from <https://www.mmu.ac.uk/library/using-the-library/policies-and-guidelines>)

Monte Carlo simulation of X-ray imaging and spectroscopy experiments using quadric geometry and variance reduction techniques.

B. Golosio^{a,c}, A. Brunetti^{a,c}, G. L. Masala^b, P. Oliva^{b,c}, T. Schoonjans^b

^aPOLCOMING Department, Section of Information Engineering, University of Sassari (Italy)

^bDipartimento di Chimica e Farmacia, University of Sassari (Italy)

^cIstituto Nazionale di Fisica Nucleare, Sezione di Cagliari (Italy)

*Corresponding author: Bruno Golosio, POLCOMING Department, Section of Information Engineering, University of Sassari, Viale Mancini 5, 07100, Sassari (Italy) Phone +39(79)229587, golosio@uniss.it

Abstract

The simulation of X-ray imaging experiments is often performed using deterministic codes, which can be relatively fast and easy to use. However, such codes are generally not suitable for the simulation of even slightly more complex experimental conditions, involving, for instance, first-order or higher-order scattering, X-ray fluorescence emissions, or more complex geometries, particularly for experiments that combine spatial resolution with spectral information. In such cases, simulations are often performed using codes based on the Monte Carlo method. In a simple Monte Carlo approach, the interaction position of an X-ray photon and the state of the photon after an interaction are obtained simply according to the theoretical probability distributions. This approach may be quite inefficient because the final channels of interest may include only a limited region of space or photons produced by a rare interaction, e.g., fluorescent emission from elements with very low concentrations. In the field of X-ray fluorescence spectroscopy, this problem has been solved by combining the Monte Carlo method with variance reduction techniques, which can reduce the computation time by several orders of magnitude. In this work, we present a C++ code for the general simulation of X-ray imaging and spectroscopy experiments, based on the application of the Monte Carlo method in combination with variance reduction techniques, with a description of sample geometry based on quadric surfaces. We describe the benefits of the object-oriented approach in terms of code maintenance, the flexibility of the program for the simulation of different experimental conditions and the possibility of easily adding new modules. Sample applications in the fields of X-ray imaging and X-ray spectroscopy are discussed.

Keywords: X-ray imaging, X-ray spectroscopy, Monte Carlo simulation

Program Summary

Program title: XRMCM version 6.3.3

Catalogue identifier:

Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland

Licensing provisions: GNU General Public License version 3 <<http://www.gnu.org/licenses/>>

No. of lines in distributed program, including test data, etc.: 227028

No. of bytes in distributed program, including test data, etc.: 2.2 MB

Distribution format: tgz

Programming language: C++

Computer: Tested on several PCs and on Mac

Operating system: Linux, Mac OS X, Windows (native and cygwin)

RAM: It is dependent on the input data but usually between 1 and 10 MB.

Classification: PACS 87.10.Rt Monte Carlo simulations, 87.59.-e X-ray imaging, 87.64.-t Spectroscopic and microscopic techniques in biophysics and medical physics, 87.64.Aa Computer simulation.

Nature of problem: Simulation of a wide range of X-ray imaging and spectroscopy experiments using different types of sources and detectors.

Solution method: XRMC is a versatile program that is useful for the simulation of a wide range of X-ray imaging and spectroscopy experiments. It enables the simulation of monochromatic and polychromatic X-ray sources, with unpolarised or partially/completely polarised radiation. Single-element detectors as well as two-dimensional pixel detectors can be used in the simulations, with several acquisition options. In the current version of the program, the sample is modelled by combining convex three-dimensional objects demarcated by quadric surfaces, such as planes, ellipsoids and cylinders. The Monte Carlo approach makes XRMC able to accurately simulate X-ray photon transport and interactions with matter up to any order of interaction. The differential cross-sections and all other quantities related to the interaction processes (photoelectric absorption, fluorescence emission, elastic and inelastic scattering) are computed using the xraylib software library, which is currently the most complete and up-to-date software library for X-ray parameters. The use of variance reduction techniques makes XRMC able to reduce the simulation time by several orders of magnitude compared to other general-purpose Monte Carlo simulation programs.

Running time: It is dependent on the complexity of the simulation. For the examples distributed with the code, it ranges from less than 1 second to a few minutes.

Introduction

Variance reduction techniques for the simulation of X-ray photon transport allow for the combination of the speed of deterministic methods and the ability of the Monte Carlo approach to simulate photon histories up to the first or higher interaction order under complex experimental conditions. Such techniques have been widely used in the field of X-ray fluorescence spectroscopy [1-5].

Some general-purpose Monte Carlo simulation codes make limited use of variance reduction techniques [6, 7], for instance by providing the option of forcing a particle to end its trajectory in a specific region of space. However, most X-ray imaging and spectroscopy applications require a greater computational effort. Expert users often incorporate other types of variance reduction techniques by modifying specific parts of general-purpose Monte Carlo codes. Therefore, the derivation and implementation of such techniques are often considered an art by Monte Carlo software developers.

Bottiglieri et al. [8] presented a code for the simulation of X-ray imaging and spectroscopy experiments performed on samples described in a three-dimensional regular grid, with elemental composition and mass density specified for each grid node (voxel). In this work, we present a program for the simulation of X-ray imaging and spectroscopy experiments based on the Monte Carlo method by exploiting variance reduction techniques, with a description of the sample geometry based on quadric surfaces. Past versions of this program have been used in medical imaging [9-13], archaeometry [14-16] and beam characterisation [17-19] with X-ray tubes as well as with less conventional X-ray sources (e.g., inverse Compton scattering sources [9-13,17,18]) for planning experimental setups, optimising experimental parameters and comparing theoretical models with experimental results. The code is written entirely in the C++ programming language. A description of the main variance reduction techniques used for the Monte Carlo simulation of X-ray photon transport can be found in ref. [1-5,8]. The present work instead focusses on a description of the advantages of an object-oriented approach for the general simulation of X-ray imaging and spectroscopy experiments and on the use of quadric surfaces for the geometrical modelling of experiments. The following sections describe the main classes used in the code with their member variables and functions.

Method

The standard experimental setup simulated by XPMC consists of an X-ray source, a sample and a detector.

The source can be a point-like source or it can have a three-dimensional Gaussian distribution. The radiation emitted by the source can be unpolarised, partially polarised or fully polarised. The energy spectrum can include a continuous component and a set of discrete energy lines, which can either be monochromatic or follow Gaussian distributions with specified standard deviations.

The sample is composed of a number of materials, called *phases*, which are characterised by their mass density and by their elemental composition, i.e., the atomic numbers and weight fractions of the atomic species that define them. Each phase is assumed to be homogeneous. The sample geometry is specified using a set of three-dimensional objects delimited by quadric surfaces.

The detector can be either a pixel array or a single-element detector. Each pixel/element can have a rectangular or an elliptical shape. It can either record the number of X-ray photons collected by each pixel over a specified exposure time, the total deposited energy of these photons or the full spectrum of the radiation detected by each pixel. The source, the sample and the detector can be placed at arbitrary positions in arbitrary orientations.

XPMC can simulate the history of individual X-ray photons, starting from the source and ending at the detector. The trajectory of a photon is modelled as consisting of a sequence of straight paths, each of which is terminated by an interaction of the photon with a sample atom. The interaction processes that can be simulated by XPMC are the photoelectric effect (eventually followed by fluorescence emission), Rayleigh (elastic) scattering and Compton (inelastic) scattering. The (differential) cross-sections for these interactions and the fluorescence line energies are computed using the *xraylib* software library [20,21]. The code (currently) does not support the transport of electrons, which can be produced by the photoelectric effect, Auger emission or Compton scattering; even though such electrons may produce secondary photons by bremsstrahlung or impact ionisation, in most practical experimental conditions their contribution is not significant.

At each step of the simulation, the state of a photon is defined by its position vector \mathbf{r}_{ph} , its direction, its polarisation vector and its energy E . Because variance reduction techniques are used, a weight w is also associated with the photon. The trajectory of the photon is considered to be straight until it is absorbed, scattered or detected. With the variance reduction techniques exploited by XPMC, the photon history never ends with absorption: at each interaction point, the photon is either scattered by a coherent or incoherent scattering process or re-emitted by fluorescence emission after photoelectric absorption, with a different energy and direction and with a proper weight that takes into account the fluorescent emission cross-section [5]. Furthermore, the photon is not allowed to exit from the sample region without being detected: it is forced either to interact with the sample or to be detected [5]. Therefore, the photon trajectory can be divided into straight paths, starting from the source and ending at the detector. Let \mathbf{r}_0 and \mathbf{k}_0 be the position and the direction of the photon at the beginning of a given step. The next interaction occurs at the point $\mathbf{r}_1 = \mathbf{r}_0 + s\mathbf{k}_0$, where s is the distance from the starting point to the next interaction position. The probability distribution for the step length s depends on the values of the linear absorption coefficient $\mu(\mathbf{r}_0 + s\mathbf{k}_0, E_0)$ along the photon trajectory at the photon energy E_0 . The probability that the next interaction occurs between s and $s+ds$ is

$$\exp\left[-\int_0^s \mu(t)dt\right] \mu(s)ds \quad (1)$$

where the exponential function represents the probability that the photon does not interact before reaching the position s (survival probability) and $\mu(s)ds$ is the probability that, if the photon reaches the position s , it interacts between s and $s+ds$. The step length s can be extracted using the inverse cumulative distribution function and a pseudo-random number generator, as described in Ref. [5], which also describes how the photon can be forced to have an interaction in the sample and how the weight associated with the event should be updated.

After the extraction of the step length, the photon is moved to the interaction position \mathbf{r}_1 , and the code selects the atomic species with which the interaction will occur and the interaction type (photoelectric+fluorescence emission, elastic scattering or inelastic scattering). The probability that the interaction occurs with an atom of type α is

$$P_{atom}^{(\alpha)} = \frac{\rho w_{\alpha} \sigma_{tot}(Z_{\alpha}, E_0)}{\mu(s, E_0)} \quad (2)$$

where ρ is the mass density of the sample at the interaction position, w_{α} is the weight fraction of the element α , $\sigma_{tot}(Z_{\alpha}, E_0)$ is the total cross-section for the interaction with an atom of atomic number Z_{α} at the incident energy E_0 and $\mu(s, E_0)$ is the linear absorption coefficient at the position s .

The probability that an interaction process of type l occurs (emission of a particular fluorescence line, elastic scattering or inelastic scattering) is

$$P_{inter}^{(l)} = \frac{\sigma_l(Z_{\alpha}, E_0)}{\sigma_{tot}(Z_{\alpha}, E_0)} \quad (3)$$

where $\sigma_l(Z_{\alpha}, E_0)$ is the cross-section for this interaction process. The atomic species that is responsible for the interaction and the interaction type/fluorescence line is selected using the procedure described in Ref. [8]. In the case of fluorescence emission, the direction of the photon after the interaction is extracted from a random distribution uniform over the entire solid angle of 4π . Clearly, the new energy will be that of the fluorescence line. In the case of elastic/inelastic scattering, the new direction is distributed according to the differential cross-sections for these processes. The extraction of the new direction, energy and polarisation is also based on the method described in Ref. [5].

The signal produced by photons that start from the source and reach the detector without interacting with the sample is called the *transmitted signal*, or *zero-order* signal.

The signal due to photons that reach the detector after a single scattering or fluorescence emission process is called the *first-order* signal. Higher-order signals are defined similarly. In our approach, the contribution of different scattering orders is evaluated separately. After the last interaction, the photon is forced to reach a specific element of the detector. The final photon position is extracted according to a random uniform distribution over the element surface. The weight associated with the event is thus updated with the method described in Ref. [8].

The following sections outline the different components of the XRM C program. For a full description of the underlying C++ classes, including member variables and methods, the reader is referred to the online manual, which is continually updated to reflect the latest development status of XRM C.

Vector and matrix operations

Many of the calculations performed by the program involve operations on three-dimensional vectors and on 3×3 or 4×4 real matrices. Therefore, three specialised classes, *vect3*, *matr3* and *matr4*, have been implemented. These classes exploit the operator-overloading feature of the C++ programming language to represent vector and matrix operations in a natural and readable manner. For instance, a change of coordinates from the local coordinate system associated with a device to the absolute coordinate system of XPMC can be expressed by the following vector operation:

$$\mathbf{r} = \mathbf{r}_0 + \mathbf{u}_i \cdot x + \mathbf{u}_j \cdot y + \mathbf{u}_k \cdot z \quad (4)$$

where \mathbf{r} is the position vector of a point relative to the absolute coordinate system, \mathbf{r}_0 is the position vector of the local coordinate system origin, \mathbf{u}_i , \mathbf{u}_j and \mathbf{u}_k are the direction vectors of the local coordinate system axis and x , y and z are the coordinates of the point relative to the local coordinate system. This operation can be expressed using the *vect3* class as follows:

$$r = r0 + ui*x + uj*y + uk*z; \quad (5)$$

where r , ui , uj and uk are objects of the class *vect3*, x , y and z belong to the C/C++ type *double* and “+” and “*” are overloaded operators of the *vect3* class.

An alternative approach is based on more general matrix and vector base classes and derive classes for specific dimensions. However, the computation time overhead of this approach is significantly high; therefore, the ultimately decided that specialised classes for three and four dimensions be used.

Several member functions were written for *vect3*, *matr3* and *matr4*, facilitating elementary mathematical operations (addition, multiplication with a scalar, etc.) on instantiated objects, as well as typical vector/matrix operations (dot and cross products, etc.).

The *photon* class

XPMC uses a specialised *photon* class to describe X-ray photon transport and interaction with matter. A local coordinate system is associated with the photon, with the direction of the three axes x , y , and z defined by three unit vectors \mathbf{i}_{ph} , \mathbf{j}_{ph} , \mathbf{k}_{ph} , where \mathbf{k}_{ph} is the photon direction, \mathbf{i}_{ph} is the polarisation vector, which is always perpendicular to the photon direction, and \mathbf{j}_{ph} , is the unit vector perpendicular to both \mathbf{i}_{ph} and \mathbf{k}_{ph} . The other variables that describe the photon state are the photon's energy E and its weight w .

Member functions that manage the displacement of the photon, as well as the calculation of the interaction cross-sections, are implemented.

The *device* and *bodydevice* classes

In XPMC, the term *device* refers to a C++ object that is created with the intention of being used by the simulation and whose parameters are loaded from the corresponding file. However, the derived classes do not necessarily represent a physical device. For instance, the phase (material) array and the sample are considered devices. The term *device* is also the name of the abstract base class from which all concrete device classes are derived. Its

member functions are related to connecting the device objects to other devices, as well as loading parameters from the input files, etc.

Another important abstract class used in XRMC, derived from the device class, is the *bodydevice* class. This class extends the device class by adding member variables and methods that handle the position and local coordinate system of a device.

The classes derived from the *bodydevice* class are the *basesource* and *detectorarray* classes. The *basesource* class is an abstract class for X-ray sources. It is the base class for all classes that can send X-ray photons to other devices, which in the current version of the code are the *source* class (which can send photons to the sample or to detector devices) and the *sample* class (which can send photons to another sample or to detector devices). In future releases of the program, other classes derived from the *basesource* class could be used to represent optical elements, such as X-ray mirrors, multilayers and X-ray lenses. The methods that are associated with the *basesource* class handle the (forced) sending of photons to the output device (detector).

On the other hand, the concrete classes that are directly derived from the abstract device class are the *composition*, *spectrum* and *geom3d* classes.

Each device can be connected to one or more input devices. Figure 1 shows schematically how the concrete devices used in a standard setup are interconnected. The classes of those devices are described in the following paragraphs.

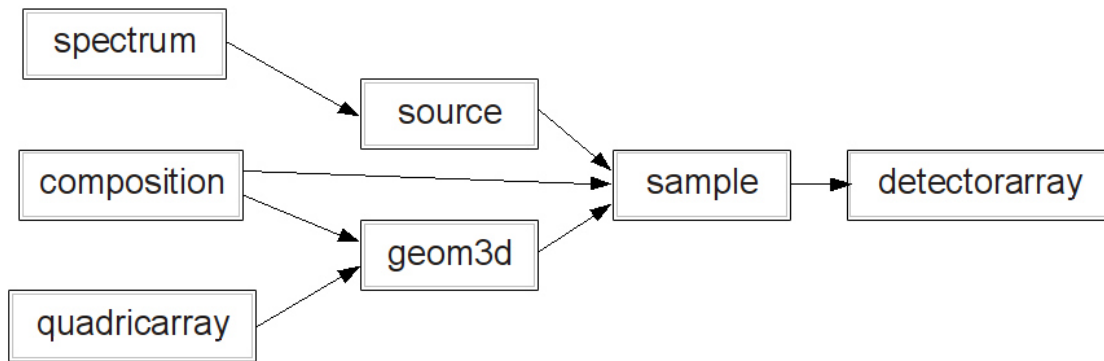


Fig. 1: C++ classes associated with the main XRMC devices and connections used in a standard setup.

The *source* class

The current version of the code assumes that the X-ray beam is produced by a point source or by an extended source with a three-dimensional Gaussian distribution. A local coordinate system is attached to the source, which is determined by the vector position of the origin \mathbf{r}_s and by the orthonormal unit vectors \mathbf{i}_s , \mathbf{j}_s and \mathbf{k}_s , which correspond to the directions of the local x , y , and z axes, respectively. The local z axis represents the main source direction, whereas the local x and y axes are used to define the beam polarisation and angular aperture. Let θ_s and ϕ_s be the polar and azimuthal angle, respectively, relative to the source coordinate system. The user can specify the angular apertures θ_x and θ_y in the x and y directions, respectively. In general, the angular aperture of the source is elliptical and is defined by the following inequality:

$$\theta^2 \leq \theta_x^2 \cos^2 \phi_s + \theta_y^2 \sin^2 \phi_s \quad (6)$$

The source intensity distribution is assumed to be uniform over the solid angle limited by this angular aperture.

The member functions connected to this class are mainly charged with calculating the initial direction and polarisation of the photon, as well as generating the local axes system.

-
-

The *spectrum* class

The energy spectrum is modelled as the sum of two components: a set of discrete lines and a continuous component. The radiation can be unpolarised, partially polarised or completely polarised. The lines can have a Gaussian or a Dirac δ distribution (the latter one being simply a particular case of Gaussian distribution, with $\sigma = 0$). Each line is specified by its mean energy E_i , by its intensity I_i and by σ_i as follows:

N_l (integer): Number of lines in the spectrum

$E_1 \quad \sigma_1 \quad I_1$ (real values): energy, width (rms) and intensity of the 1st line

.....

$E_{N_l} \quad \sigma_{N_l} \quad I_{N_l}$ (real values): energy, width (rms) and intensity of the N^{th} line

In the case of (partially or completely) polarised radiation, the intensities of the two components polarised along the local x and y directions are specified separately for each line as follows:

$E_1 \quad \sigma_1 \quad I_{x1} \quad I_{y1}$ (real values):

energy, width (rms) and intensities of the two polarisation components of the

1st line

.....

$E_{N_l} \quad \sigma_{N_l} \quad I_{xN_l} \quad I_{yN_l}$ (real values):

energy, width (rms) and intensities of the two polarisation components of the

N^{th} line

The continuous component is defined by N samples at arbitrary energies E_1, \dots, E_N , by specifying for each sample the corresponding height of the spectral distribution I_1, \dots, I_N :

N_c (integer): Number of sampling points in the continuous spectrum;

$E_1 \quad I_1$ (real values): energy and intensity of the 1st sampling point

.....

$E_{N_c} \quad I_{N_c}$ (real values): energy and intensity of the N^{th} sampling point;

The height of the distribution in the interval between two consecutive energies of the sample E_i, E_{i+1} is approximated by a linear function of E that ranges from I_i to I_{i+1} ; therefore, the spectrum in each interval between two consecutive samples has a trapezoidal shape. The area of the trapezium $(E_{i+1} - E_i) \cdot (I_i + I_{i+1})/2$ represents the intensity of the interval.

In the case of (partially or completely) polarised radiation, the heights of the x and y components are specified separately for each sample of the continuous component as follows:

$E_1 \quad I_{x1} \quad I_{y1}$ (real values):

energy and intensities of the two polarisation components of the 1st sampling point;

$E_{NI} \quad I_{xNI} \quad I_{yNI}$ (real values):

energy and intensities of the two polarisation components of the Nth sampling point;

There are two possible ways of extracting the initial energy of X-ray photons produced by the source:

- 1) extract random energies from the whole spectrum: the initial photon energy is extracted using the whole spectrum itself as a probability distribution;
- 2) loop over all lines and all intervals of the spectrum: a loop is made over all lines and all intervals of the spectrum; the initial energy of the photon is extracted according to the probability distribution limited to a single line or to a single interval; an event is assigned a weight proportional to the line/interval intensity.

The first method is the traditional Monte Carlo approach. Lines or regions of the spectrum with lower intensity are less represented in the generated distribution, no matter how important their contribution to the detected signal is. There may be some drawbacks to this approach. For instance, if the spectrum has a relatively low intensity at higher energies and if the sample is a strongly absorbing object, the frequency of events with higher energies will be low even though they provide the most important contribution to the detected signal.

The second approach is more similar to deterministic integration methods and should often be preferred to the first approach. All lines and all interval are equally represented in the generated statistics, and their relative probability is corrected for by using the method of event weighting.

If the second method is chosen, then for each interval of the continuous component, the software offers two possible ways of extracting the photon energy:

- 1) extract the energy randomly according to the probability distribution inside the interval itself (which is modelled by a linear function, as discussed previously);
- 2) force the photon energy to be equal to the central energy of the interval.

The second method is the *pure deterministic* approach, which should not be used under normal circumstances.

The program offers the opportunity to resample the continuous component after it is defined. In this case, the user must specify the starting energy, the energy step and the number of points used for the resampling. The intensities I_i are then recalculated for the new values of E_i . Typically, this option will not be used; however, it could be useful for variance reduction if the space between the sampling energies in the continuous component definition is too large and a finer separation is desired or, in the opposite case, if the energy step in the initial definition of the spectrum is unnecessarily small.

All member functions of this class are associated with the extraction of energies and the corresponding intensities from the discrete and/or continuous parts of the excitation spectrum.

The *phase* and *composition* classes

The sample is composed of a number of materials called *phases*. Each phase is assumed to be homogeneous. Each phase is characterised by the number of atomic species that define it, through a list of the atomic numbers and weight fractions of these species, and by its mass density.

The member functions are therefore mostly concerned with the selection of the atomic type with which the next interaction will occur, as well as the computation of absorption coefficients.

The *composition* class contains an array of the phases used by the simulation. Its only member function serves to calculate the absorption coefficients of each phase.

The *quadric*, *quadricarray* and *qvolume* classes

Sample geometry is described through a set of quadric surfaces, which are used to define the surfaces of solid objects. A quadric is a surface in three-dimensional space defined as the locus of zeros of a quadratic polynomial. A quadric is generally defined by the following algebraic equation:

$$\sum_{i,j=1}^3 x_i Q_{ij} x_j + \sum_{i=1}^3 P_i x_i + R = 0. \quad (7)$$

If we define $x_0 = 1$, then the quadric may be compactly written in vector and matrix notation as follows:

$$xAx^T = 0 \quad (8)$$

where $x = (x_0, x_1, x_2, x_3)$ is a row vector, x^T is the transpose of x (a column vector) and A is a 4x4 matrix with $A_{ij} = Q_{ij} \quad \forall i, j = 1, \dots, 3$, $A_{0i} = A_{i0} = P_i$ and $A_{00} = R$.

The matrix A must be symmetric; thus, $A_{ij} = A_{ji} \quad \forall i, j = 0, \dots, 3$.

A quadric divides the space into two regions: one with $xAx^T > 0$ and the other with $xAx^T < 0$.

We will call these two regions the space outside the quadric (or external space) and the space inside the quadric (or internal space), respectively.

Whenever a unit vector normal to the quadric surface has to be defined, by default we will assume that it is oriented toward the external space. Certain types of standard quadrics can be easily defined:

- an infinite plane, defined by the coordinates of an arbitrary point of the plane x_0, y_0, z_0 and by the unit vector perpendicular to the plane u_x, u_y, u_z ;
- an ellipsoid with semi-principal axes parallel to the x, y and z axes, defined by the centre coordinates x_0, y_0, z_0 and by the length of the semi-principal axes a, b, c ;
- a cylinder with its axis parallel to the x axis and an elliptical section, defined by the y, z coordinates of its axis y_0, z_0 and by the semiaxes of the ellipse r_y, r_z ;
- a cylinder with its axis parallel to the y axis and an elliptical section, defined by the

- x, z coordinates of its axis x_0, z_0 and by the semiaxes of the ellipse r_x, r_z ;
- a cylinder with its axis parallel to the z axis and an elliptical section, defined by the x, y coordinates of its axis x_0, y_0 and by the semiaxes of the ellipse r_x, r_y ;

In the most general case, the quadric can be defined by the coefficients A_{ij} .

Several member functions were designed for this class, in most cases to handle the construction of the geometrical shapes.

-
-

The *quadricarray* class contains an array of the quadrics used by the simulation.

A solid *object* is defined as a solid shape demarcated by a set of quadric surfaces that separates the space inside the object from the space outside it. The quadrics limiting an object must be properly oriented such that their normal vectors are directed outward with respect to the object itself.

In the current version of the implementation, objects must be convex: the user is expected to handle the splitting of non-convex objects into convex ones when using them in the simulation.

An object may contain other objects, or it may be contained in another object, as long as their delimiting surfaces are not in contact.

The class used in XPMC to represent three-dimensional objects is called *qvolume*. It contains only one member function, which is responsible for the calculation of the intersections of the photon trajectory with the defined objects.

The *geom3d* class

The *geom3d* class contains an array of the three-dimensional objects used in the simulation. As shown in Fig. 1, the class essentially represents the synthesis of the quadrics (through the *qvolumes*) and the compositions.

The *path* class

When a photon exits from the source or when it is produced by a scattering/fluorescence emission process, it follows a straight trajectory defined by its position vector \mathbf{r}_{ph} and by its direction vector \mathbf{u}_{ph} . The program evaluates the intersection of this trajectory with the quadric surfaces demarcating the objects and divides it into N_s steps with uniform phases. Each step is a segment of the trajectory delimited by its intersection with different objects. The path class holds information about the intersection of a trajectory with the quadrics and about the segments between consecutive intersections. Its member functions are responsible for the calculation of the next interaction position.

The *sample* class

The *sample* class is a container used to combine information about the sample's

composition, the sample's geometry and the type of interactions that can occur in the sample. It therefore contains references to the source, the three-dimensional model, the compositions, as well as information about the possible intersections of the photon trajectories with the objects.

The *detectorarray* class

In general, the program can simulate two-dimensional array detectors with energy binning for each pixel. A single-element detector can be simulated as a special case of an array detector with only one pixel. The pixel shape can be defined as rectangular or elliptical. The latter is particularly useful when a round, single-element detector must be simulated. A local coordinate system is associated with the detector, specified by the vector position of its geometric centre \mathbf{r}_d and by the orthonormal unit vectors \mathbf{i}_d , \mathbf{j}_d and \mathbf{k}_d (see Fig. 2), which are the directions of the local x, y and z axes, respectively. The local z axis is perpendicular to the detector surface, whereas the local x and y axes are parallel to the detector rows and columns, respectively.

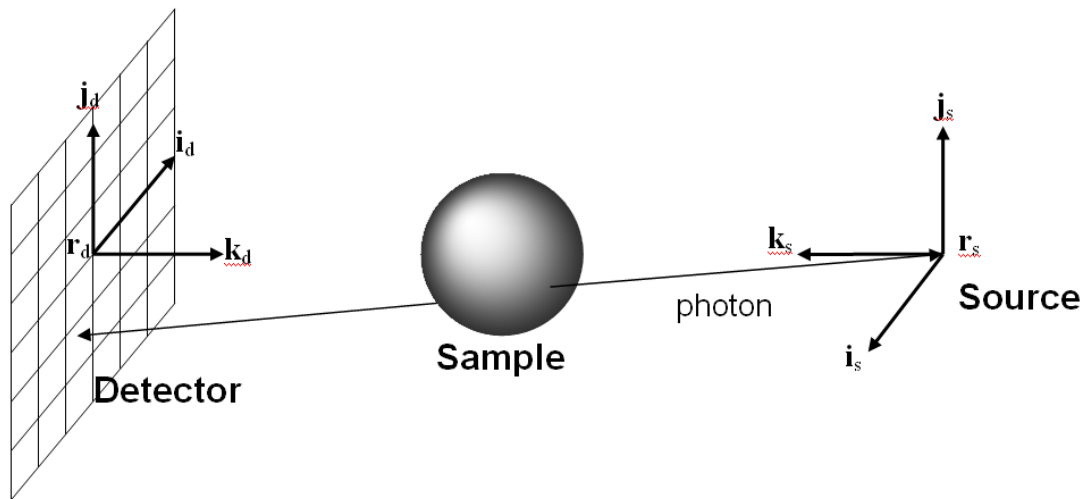


Fig. 2: The standard experimental setup simulated by XRMC consists of an X-ray source, a sample and a detector (two-dimensional array or single element). A local coordinate system is attached to the source and to the detector. \mathbf{r}_s and \mathbf{r}_d are the position vectors of the source and of the detector geometric centre, respectively. \mathbf{i}_s , \mathbf{j}_s and \mathbf{k}_s are the directions of the source local x, y and z axes, respectively. \mathbf{i}_d , \mathbf{j}_d and \mathbf{k}_d are the directions of the detector local x, y and z axes, respectively. The detector local z axis is perpendicular to the detector surface, whereas the local x and y axes are parallel to the detector rows and columns, respectively.

Two acquisition modalities are possible:

- 1) fluence: each channel simply counts the number of photons that it detects;
- 2) energy fluence: each channel sums up the energy of the photons that it detects.

The energy response of the detector can eventually be taken into account by using the first modality with a sufficient number of energy bins and by a proper post-processing of the acquisition. The member functions of the *detectorarray* class are responsible for initiating the acquisition (and therefore the actual simulation) and for clearing the contents of the virtual detector channels.

•

Results

The code was verified by experimental measurements and, whenever possible, against theoretical predictions for a wide range of experimental conditions for imaging and for spectroscopy.

An example is schematically depicted in Fig. 3, which represents a setup used to test the simulation of second-order scattering against theoretical predictions.

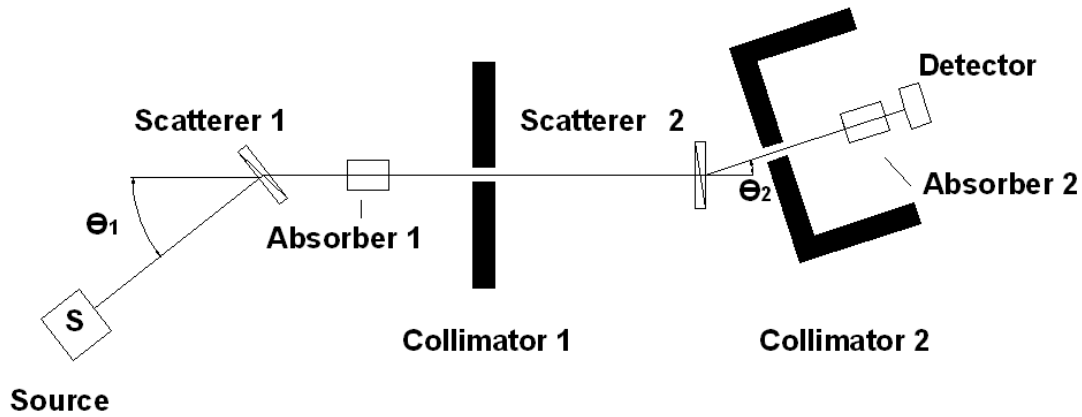


Fig. 3: Schematic representation of the setup used to test the simulation of second-order scattering against theoretical predictions.

All objects are cylinders (or cylindrical shells), with the main axis lying on the figure plane. This setup was designed such that the only relevant contribution to the second-order scattering signal on the detector is due to a first scattering process on the object *scatterer 1* and a second scattering process on the object *scatterer 2*. The two absorbers are also used to test the simulation results against theoretical predictions; although they contribute to the scattering signal on the detector, this contribution is relevant only for scattering orders higher than the second order. The radiation produced by the source is completely polarised in the direction perpendicular to the figure plane. The two scattering objects and the detector are not shown to scale in the figure. The relevant input parameters of the simulation are as follows:

- monochromatic source with $E_0=50$ keV;
- scatterer 1 thickness: $s_s=0.1$ mm, diameter: $d_1=2$ cm, material: aluminium (density $\rho=2.7$ g/cm³);
- scatterer 2 thickness: $s_s=0.1$ mm, diameter: $d_2=4$ mm, material: aluminium;
- absorber 1 and 2 thickness: $s_a=2$ cm, material: polymethyl methacrylate (PMMA);
- collimator 1 and 2 thickness: 2 cm, hole diameter: 4 mm, material: lead;
- detector diameter: $d_d=4$ mm
- scatterer 1 – scatterer 2 distance: $l_1=20$ cm;
- scatterer 2 – detector distance: $l_2=10$ cm.

The probability that a photon produced by the source is scattered from the object *scatterer 1* and reaches the object *scatterer 2* is

$$P_1 = \rho_s \Omega_1 \frac{d\sigma_A}{d\Omega} \exp(-\mu_1 s_a) \quad (9)$$

where A denotes the first scattering process type (elastic or inelastic), $\Omega_1 = \pi d_2^2 / 4l_1^2$ is the solid angle from the first interaction point to the surface of scatterer 2, $d\sigma_A / d\Omega$ is the differential cross-section for the first scattering interaction at incident energy E_0 and scattering polar angle θ_1 and μ_1 is the PMMA absorption coefficient at energy E_1 after the scattering interaction. Clearly, for elastic scattering $E_1=E_0$, whereas for inelastic scattering, the energy can be calculated by the well-known Compton formula.

The probability that a photon incident on *scatterer 2* is scattered and reaches the detector is

$$P_2 = \rho_s \Omega_2 \frac{d\sigma_B}{d\Omega} \exp(-\mu_2 s_a) \quad (10)$$

where B denotes the second scattering process type (elastic or inelastic), $\Omega_2 = \pi d_d^2 / 4l_2^2$ is the solid angle from the second interaction point to the detector surface, $d\sigma_B / d\Omega$ is the differential cross-section for the second scattering interaction at incident energy E_1 and scattering polar angle θ_2 and μ_2 is the PMMA absorption coefficient at the energy E_2 after the second scattering interaction.

The total probability P_{tot} is the product $P_1 P_2$ evaluated and summed for all combinations of the two scattering types A - B , i.e., elastic-elastic, elastic-inelastic, inelastic-elastic and inelastic-inelastic. P_{tot} is on the order of 10^{-14} . A simulation based on the conventional Monte Carlo approach would not be able to produce sufficient statistics regarding detected photons within an acceptable time frame. Through the variance reduction techniques implemented in XRMCA, the fraction of the (weighted) events that yield a useful signal on the detector is on the order of $5 \cdot 10^{-5}$, i.e., it is more than nine orders of magnitude greater than that of the conventional Monte Carlo approach; therefore, it is possible to collect a good precision for given values of θ_1 and θ_2 within a few hours of simulation. The photon count on the detector is calculated as follows:

$$C = IP_{\text{tot}} t \quad (11)$$

where I is the source intensity and t is the exposure time. Figure 4 shows a comparison between the theoretical curve of C as a function of θ_2 and values obtained from simulations, for $\theta_1=45^\circ$, $I=10^{10}$ ph/s and $t=1$ s. The point $\theta_2=0$ is excluded from the simulation because for that point the scattering from both collimators' hole surfaces contribute significantly to the second-order scattering signal. The drop close to 90 degrees in the simulation data is related to the self-absorption of the *scatterer 2*, which becomes relevant at such angles; this effect was not taken into account in the theoretical model because it depends not only on the angle θ_2 but also on the second interaction position, which is randomly distributed throughout the object volume.

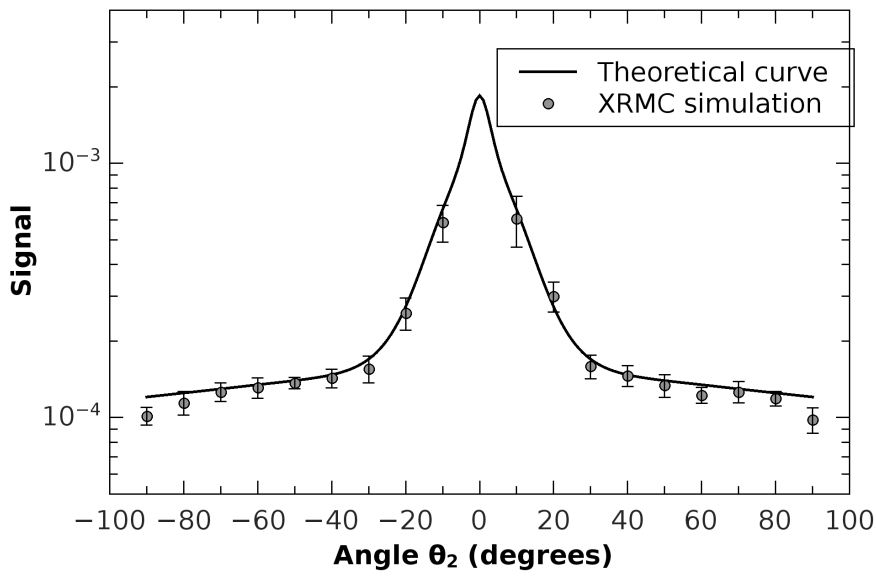
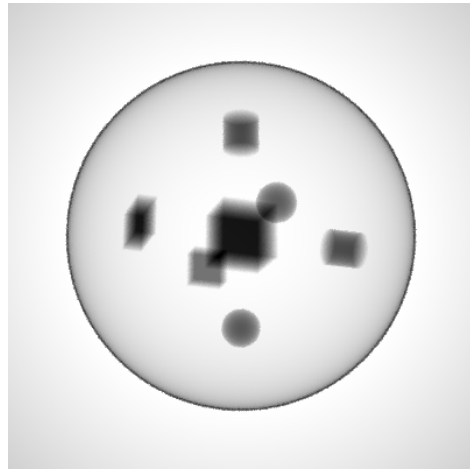


Fig. 4. Comparison between the theoretical curve for the second-order scattering process shown in Fig. 3 and the signal obtained by XPMC simulations.



(a)



(b)



(c)



(d)

Fig. 5: Example of simulated radiographic images of different samples: a star shape (a), a thin spherical shell containing objects delimited by different type of quadrics (b), a wheel shape (c) and a cylindrical shell (d).

Possible applications of XRM C in X-ray imaging are described in the tutorial, which is distributed with the program. The corresponding input files can be found in the *example* directory. These examples explain, through comments in the input files, how to build the experimental setup, how to define the elemental compositions and how to define the object geometry using different types of quadrics and also the rotation/translation functionality. A selection of images resulting from those examples are shown in Fig. 5.

XRM C can also be applied to simulate realistic X-ray fluorescence spectroscopy measurements, as demonstrated in the tutorial. Fig. 6 presents a schematic of the experimental setup.

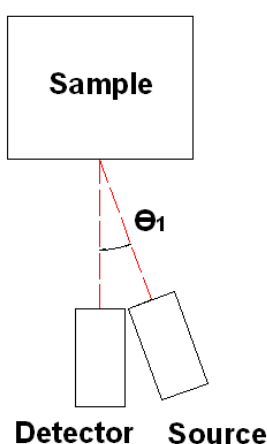


Fig. 6: Schematic representation of the setup used for fluorescence spectroscopy measurements and simulations.

The source is a Ag-anode X-ray tube operated at 40 kV and 10 μ A. The source spectrum is polychromatic and was determined experimentally. The beam was collimated through a 1-mm-wide cylindrical collimator. The detector is a silicon drift detector (X-123SDD by Amptek) measuring 25 mm² in diameter and 500 μ m in thickness. The sample is a parallelepiped composed of the following three layers (in order of increasing distance to the source):

- a simulated 200- μ m-thick surface mixture layer;
- a 300- μ m-thick copper layer;
- a 2.5-mm-thick mixture layer similar to the first one.

Similar samples are often encountered in cultural heritage research, such as in studies of excavated ancient metals, in which case the surface layers are due to interactions with sediment. The mixture is a seven-element phase, whose composition can be found in the *composition.dat* file in the *example/fluor_layers* directory.

The current version of XRM C does not include the simulation of the response of real detectors. The users are expected to perform any post-processing of the simulation output. With respect to our example, a separate program that allows for the simulation of the

response of the silicon drift detector is included. This program can be found in the directory `example/fluor_layers/src`, where a README file explains how to compile and run the utility. Figure 7 shows a comparison between the measured signal and the simulated one. In the experimental spectrum, there are peaks due to pile-up effects and X-ray tube emission that are not present in the simulations.

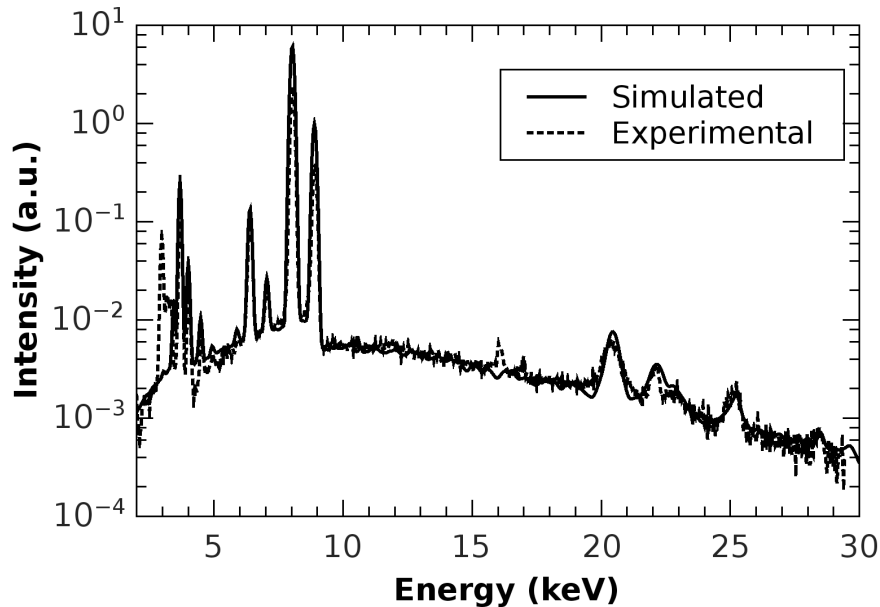


Fig. 7: Comparison between the simulated and measured spectra for a fluorescence spectroscopy experiment.

Conclusion

XRMC is a versatile program that is useful for the simulation of a wide range of X-ray imaging and spectroscopy experiments. The program allows for the simulation of monochromatic and polychromatic X-ray sources with unpolarised or partially/completely polarised radiation. Single-element detectors as well as two-dimensional pixel detectors can be used in the simulations, with several acquisition options available. In the current version of the program, the sample is modelled by combining convex three-dimensional objects demarcated by quadric surfaces, such as planes, ellipsoids and cylinders. The Monte Carlo approach enables XRMC to accurately simulate X-ray photon transport and interactions with matter up to any order of interaction. The differential cross-sections and all other quantities related to the interaction processes (photoelectric absorption, elastic and inelastic scattering) and to fluorescence emission are computed using the xraylib software library [20,21], which is currently the most complete and up-to-date software library for X-ray parameters. The use of variance reduction techniques enables XRMC to reduce the simulation time by several orders of magnitude compared to other general-purpose Monte Carlo simulation programs. It should be noted that the program XMI-MSIM [1-5] is currently superior to XRMC in simulating X-ray fluorescence (XRF) experiments. However, XMI-MSIM can only simulate samples composed of parallel layers. For X-ray imaging experiments, for experiments that combine imaging and spectroscopy or for more complex sample geometries, XRMC is currently one of the most valuable programs available.

The use of an object-oriented approach, incorporating the concept of devices as independent computational modules that can be connected to each other to construct an experimental setup, make XRMC a flexible framework into which new modules can be

easily integrated. Current work is focussed on the implementation of the response of realistic detectors for imaging (e.g., CMOS detectors) and for spectroscopy (e.g., Silicon Drift detectors) as devices that can be used within the program. New modules will soon be included in the program for a voxel-based description of sample geometry and for phase contrast imaging simulations [19].

XRMC is released under the GNU general public license and can be downloaded freely from <http://github.com/golosio/xrmc>. Participation in the development is possible and encouraged through the Github website.

Acknowledgements

This work has been supported by Regione Sardegna, "progetto di ricerca di base Legge 7/2007, Bando 2008, Studio e realizzazione di sistemi di calcolo scientifico paralleli a basso costo basati su processori grafici, CUP: J71J10000070002".

References

- [1] L. Vincze, K. Janssens, F. Adams, A general Monte Carlo simulation of energy-dispersive X-ray fluorescence spectrometers: I. Unpolarized radiation, homogeneous samples, *Spectrochim. Acta Part B* 48 (4) (1993) 553 – 573.
- [2] L. Vincze, K. Janssens, F. Adams, L. Rivers, W. Jones, A general Monte Carlo simulation of ED-XRF spectrometers: II. Polarized monochromatic radiation, homogeneous samples, *Spectrochim. Acta Part B* 50 (2) (1995) 127 – 147.
- [3] L. Vincze, K. Janssens, F. Adams, W. Jones, A general Monte Carlo simulation of energy-dispersive X-ray fluorescence spectrometers: III. Polarized polychromatic radiation, homogeneous samples, *Spectrochim. Acta Part B* 50 (1995) 1481 – 1500.
- [4] L. Vincze, K. Janssens, B. Vekemans, F. Adams, Monte Carlo simulation of X-ray fluorescence spectra: Part 4. Photon scattering at high X-ray energies, *Spectrochim. Acta Part B* 54 (1999) 1711 – 1722.
- [5] T. Schoonjans, L. Vincze, V.A. Solé, M. Sanchez del Rio, P. Brondeel, G. Silversmit, K. Appel, C. Ferrero, A general Monte Carlo simulation of energy-dispersive X-ray fluorescence spectrometers — Part 5. Polarized radiation, stratified samples, cascade effects, M-lines, *Spectrochim. Acta Part B* 70 (2012) 10–23.
- [6] X. Llovet, J. Fernandez-Varea, J. Sempau, F. Salvat, Monte Carlo simulation of X-ray emission using the general-purpose code PENELOPE, *Surf. Interface Anal.* 37 (2005) 1054–1058.
- [7] J. Hendricks, K. Adams, T. Booth, J. Briesmeister, L. Carter, L. Cox, J. Favorite, R. Forster, G. McKinney, R. Prael, Present and future capabilities of MCNP, *Appl. Radiat. Isot.* 53 (2000) 857–861.
- [8] U. Bottigli, A. Brunetti, B. Golosio, P. Oliva, S. Stumbo, L. Vincze, P. Randaccio, P. Bleuett, A. Simionovici, A. Somogyi, Voxel-based Monte Carlo simulation of X-ray imaging and spectroscopy experiments, *Spectrochim. Acta Part B* 59 (2004) 1747–1754. [9] P. Oliva, B. Golosio, S. Stumbo, A. Bravin, P. Tomassini, Compact x-ray sources for mammographic applications: Monte Carlo simulations of image quality, *Medical Physics*, 36 (11) (2009) 5149-5161.
- [10] P. Oliva, A. Bacci, U. Bottigli, M. Carpinelli, P. Delogu, M. Ferrario, D. Giuliotti, B. Golosio, V. Petrillo, L. Serafini, P. Tomassini, C. Vaccarezza, C. Vicario, A. Stefanini, Start-to-end simulation of a Thomson source for mammography, *Nuclear Instruments and Methods A* 615 (1), (2010) 93-99.
- [11] U. Bottigli, B. Golosio, G.L. Masala, P. Oliva, S. Stumbo, A. Bravin, A. Bacci, L. Serafini, C. Maroli, V. Petrillo, M. Ferrario, C. Vaccarezza, Effect of different spectral distributions to image a contrast detail phantom in the mammography energy range, *Nuovo Cimento della Societa Italiana di Fisica C* 29(2) (2006) 215-228.
- [12] P. Oliva, B. Golosio, S. Stumbo, M. Carpinelli, Advantages of quasi-monochromatic X-ray sources in absorption mammography, *Nuclear Instruments and Methods A* 608 (2009) S106-S108.
- [13] B. Golosio, P. Delogu, I. Zanette, P. Oliva, A. Stefanini, G. Stegel, M. Carpinelli, Visibility of tumor-like details in inline phase contrast mammography using quasimonochromatic X-ray

sources,

Nuclear Instruments and Methods A 608 (2009) S66-S69.

[14] G. Piga, A. Santos-Cubedo, S. Moya Solà, A. Brunetti, A. Malgosa, S. Enzo, An X-ray Diffraction (XRD) and X-ray Fluorescence (XRF) investigation in human and animal fossil bones from Holocene to Middle Triassic, *Journal of Archaeological Science* 36 (2009) 1857-1868.

[15] G. Piga, A. Santos-Cubedo, A. Brunetti, M. Piccinini, A. Malgosa, E. Napolitano, S. Enzo, A multi-technique approach by XRD, XRF, FT-IR to characterize the diagenesis of dinosaur bones from Spain, *Palaeogeography, Palaeoclimatology, Palaeoecology* 310(1-2) (2011) 92-107.

[16] G. Piga, G. Solinas, T.J.U. Thompson, A. Brunetti, A. Malgosa, S. Enzo, Is X-ray diffraction able to distinguish between animal and human bones?, *Journal of Archaeological Science* 40 (2013) 778-785.

[17] B. Golosio, M. Endrizzi, P. Oliva, P. Delogu, M. Carpinelli, I. Pogorelsky, V. Yakimenko, Measurement of an inverse Compton scattering source local spectrum using k-edge filters, *Applied Physics Letters*, 100 (16) (2012) art. no. 164104.

[18] P. Oliva, M. Carpinelli, B. Golosio, P. Delogu, M. Endrizzi, J. Park, I. Pogorelsky, V. Yakimenko, O. Williams, J. Rosenzweig, Quantitative evaluation of single-shot inline phase contrast imaging using an inverse Compton x-ray source, *Applied Physics Letters* 97 (13) (2010) art. no. 134104.

[19] B. Golosio, P. Delogu, I. Zanette, M. Carpinelli, G.L. Masala, P. Oliva, A. Stefanini, S. Stumbo, Phase contrast imaging simulation and measurements using polychromatic sources with small source-object distances, *Journal of Applied Physics* 104(9) (2008) art. no. 093102.

[20] A. Brunetti, M. Sanchez del Rio, B. Golosio, A. Simionovici, A. Somogyi, A library for X-ray-matter interaction cross sections for X-ray fluorescence applications, *Spectrochim. Acta Part B* 59 (2004) 1725-1731.

[21] T. Schoonjans, A. Brunetti, B. Golosio, M. Sanchez del Rio, V.A. Solé, C. Ferrero, L. Vincze, The xraylib library for X-ray-matter interactions. Recent developments, *Spectrochim. Acta Part B* 66 (2011) 776-784.