The definitive version is available at `http://www.sciencedirect.com/`.

F. P. Vidal, and P.-F. Villard: Development and Validation of Real-time Simulation of X-ray Imaging with Respiratory Motion. In *Computerized Medical Imaging and Graphics*. Volume 49, April 2016, Pages 1–15. Elsevier.

DOI: 10.1016/j.compmedimag.2015.12.002

Keywords: X-ray simulation; deterministic simulation (ray-tracing); digitally reconstructed radiograph; respiration simulation; medical virtual environment; imaging guidance; interventional radiology training.

PACS: 87.59.-e; 87.19.Wx; 07.05.Tp; 02.50.Ng; 42.15.Dp.

```
@article{Vidal2016ComputMedImagingGraph,
  author = {F. P. Vidal and {P.-F.} Villard},
  title = {Development and Validation of Real-time Simulation of X-ray Imaging
    with Respiratory Motion},
  journal = {Computerized Medical Imaging and Graphics},
  year = 2016,
  volume = 49,
  pages = {1--15},
  month = apr,
  abstract = {We present a framework that combines evolutionary optimisation,
    soft tissue modelling and ray tracing on GPU to simultaneously compute
    the respiratory motion and X-ray imaging in real-time. Our aim is
    to provide validated building blocks with high fidelity to closely match
    both the human physiology and the physics of X-rays. A CPU-based
    set of algorithms is presented to model organ behaviours during respiration.
    Soft tissue deformation is computed with an extension of the Chain Mail
    method. Rigid elements move according to kinematic laws.
    A GPU-based surface rendering method is proposed to compute
    the X-ray image using the Beer-Lambert law. It is provided as
    an open-source library. A quantitative validation study is provided
    to objectively assess the accuracy of both components:
    i) the respiration against anatomical data, and
    ii) the X-ray against the Beer-Lambert law and the results of Monte Carlo
    simulations.
    Our implementation can be used in various applications, such as interactive
    medical virtual environment to train percutaneous transhepatic
    cholangiography in interventional radiology, 2D/3D registration, computation
    of digitally reconstructed radiograph, simulation of 4D sinograms to test
    tomography reconstruction tools.},
  pmid = {to appear},
  publisher = {Elsevier}
}
```

Our demos are available on the website of the gVirtualXRay project at `http://gvirtualxray.sourceforge.net/`

Development and Validation of Real-time Simulation of X-ray Imaging with Respiratory Motion

F. P. Vidal [1,2], and P.-F. Villard [3,4,5]

[1] School of Computer Science, Bangor University, LL57 1UT, United Kingdom

[2] Research Institute of Visual Computing, RIVIC, United Kingdom

[3] Université de Lorraine, LORIA, UMR 7503, Vandoeuvre-lès-Nancy F-54506, France

[4] Inria, Villers-lès-Nancy F-54600, France

[5] CNRS, LORIA, UMR 7503, Vandoeuvre-lès-Nancy F-54506, France

Abstract

We present a framework that combines evolutionary optimisation, soft tissue modelling and ray tracing on GPU to simultaneously compute the respiratory motion and X-ray imaging in real-time. Our aim is to provide validated building blocks with high fidelity to closely match both the human physiology and the physics of X-rays. A CPU-based set of algorithms is presented to model organ behaviours during respiration. Soft tissue deformation is computed with an extension of the Chain Mail method. Rigid elements move according to kinematic laws. A GPU-based surface rendering method is proposed to compute the X-ray image using the Beer-Lambert law. It is provided as an open-source library. A quantitative validation study is provided to objectively assess the accuracy of both components: i) the respiration against anatomical data, and ii) the X-ray against the Beer-Lambert law and the results of Monte Carlo simulations. Our implementation can be used in various applications, such as interactive medical virtual environment to train percutaneous transhepatic cholangiography in interventional radiology, 2D/3D registration, computation of digitally reconstructed radiographs, simulation of 4D sinograms to test tomography reconstruction tools.

Keywords: X-ray simulation; deterministic simulation (ray-tracing); digitally reconstructed radiograph; respiration simulation; medical virtual environment; imaging guidance; interventional radiology training..

PACS: 87.59.-e; 87.19.Wx; 07.05.Tp; 02.50.Ng; 42.15.Dp.

1 Introduction

There is a growing need for fast, accurate and validated tools for the virtual physiological human (VPH) in physics, imaging, and simulation in medicine. The aim of VPH is to provide a digital model of the human physiology as a single complex system. This research's contribution is twofold: provide respiration modelling with real-time performance, generate accurate X-ray images from the virtual patient. The output can be exploited in many different contexts.

Researchers in bio-medical engineering have been working on these topics for some time. However, the implementation of such models are usually not publicly available. It makes it difficult to re-use them in medical applications, or compare new models with them. In this paper, we describe a C++ implementation of the respiration model and an OpenGL implementation of our X-ray simulation code. This component is now mature and it is written as a stand-alone library: *gVirtualXRay*. We have opened its source code to the public under the BSD open-source license (permitting free reuse by academia and industry) and it is available at `http://gvirtualxray.sourceforge.net`. We extensively compare it with a state-of-the-art Monte Carlo simulation tool used in nuclear physics.

Our code can be used in virtual environments (VEs) designed for training invasive medical procedures such as interventional radiology (IR) [9] where real-time interactivity and numerical accuracy are both essential and cannot be compromised. VE-based simulators are more and more accepted for surgical training [29]. A few commercial and academic solutions have been produced in recent years. They include most of the time i) haptic component, ii) performance metrics and iii) graphic rendering. Current virtual simulator areas include endoscopic surgery [35], laparoscopic simulator [20], arthroscopic knee simulator [8] and liver biopsy interventional radiology [13]. They often overlook respiratory motion of the virtual patient's anatomy. This paper addresses the need to provide support for respiratory motion in simulated percutaneous transhepatic cholangiography (PTC). This IR procedure uses fluoroscopy (real-time X-ray imaging) to track a needle as it is inserted, during breath-hold, deeply into the liver. Once the needle has attained a sufficient depth (10-12cms), the patient is asked to breathe shallowly. X-ray-opaque (contrast) medium is gently injected through the needle as it is slowly withdrawn, observing for characteristic visual appearances of contrast entering a bile duct, whereupon, needle withdrawal is stopped. At this point, further access techniques can introduce a catheter, or therapeutic interventional instruments, into the bile duct system.

The intrinsic motion of internal anatomical structures presents significant challenges to accurate, image guided, needle placement. In interpreting abdominal respiratory organ motion (external), the operator obtains few cues from the skin surface; yet simulating visceral respiratory motion (internal) is highly complex and the computing time constraint has to be taken into account in order to be incorporated into a VE. The motion of internal structures can also be monitored using fluoroscopy. Note that fluoroscopy is not used continuously to reduce the radiation dose received by both the patient and clinician.

Simulation of X-ray imaging is important in physics, with applications in medicine, crystallography, astronomy and nondestructive testing, yet has been largely overlooked by the computer graphics community. X-ray simulation in our context is essential to a range of medical simulations that require an interactive response to match the acquisition time of a real fluoroscopy system (25-30 Hz). We take advantage of recent developments in computer graphics hardware to achieve an accelerated simulation of X-ray attenuation calculated using the Beer-Lambert law [37].

Our framework can also be used as a building block to solve the inverse problem of non-rigid registration. Digitally reconstructed radiographs (DRRs) around a 3D volume dataset are generated (a DRR is a 2D X-ray image computed from a 3D computed tomography (CT) dataset). A "model" is deformed so that the error between its own X-ray projections and the DRRs is as small as possible. It is often solved iteratively, which means that many intermediate images are computed.

Another application is in medical physics, particularly CT reconstruction. Patient motion (including internal motions such as respiration) can cause blurring, ghost images and long range streaks [7]. The simulation framework can be used to create simulated sinograms (a sinogram is the raw data produced by CT scanners prior to tomography reconstruction) of realistic controlled test-cases [36]. The respiration can be added to provide 4D data (i.e. 3D + time) to illustrate its effect on tomography reconstruction, including the assessment of respiration motion compensation techniques in low dose cone-beam computed tomography (CBCT).

Part of this research has been previously published. [40] and [41] focused on clinical value of the respiration model rather than the scientific aspect of the work. No technical detail and no quantitative validation were included. Here, we address all these deficiencies. Particularly, we completely describe the Chainmail implementation including a study of the induced soft-tissue behaviour, an analysis of the parameters influence and a complete description of how the organs are tethered all together. The way to parameterize the respiration model as an optimization problem using evolutionary computing was published in [39]. The focus of the validation was to demonstrate the superiority of our *ad-hoc* optimization framework over more traditional black-box optimization tools. X-ray simulation was initially published in [37]. It was limited to the monochormatic case, one infinitely small point source, and the X-ray beam had to be perpendicular to the detector. The initial code was validated against a private library therefore results were not reproducible. Polychromatism and geometrical unsharpness were introduced in [38]. However, no technical detail

and no validation were included. Here, we add parallel X-ray beams, the possibility to place and orientate the X-ray detector regardless of the direction of the X-ray beam (they do not have to be perpendicular to each other), the properties of human tissues with respect to X-rays are model accurately, taking into account their density and atomic elements. For transparency purposes, the validation tests and data are reproducible and publicly available on the project website.

The paper provides a detailed overview of the implementation and quantitative validation of the two main software components of our simulation framework. It can help to improve the realism of virtual reality (VR) simulations. The following sections describe related work, our techniques to compute respiratory motion and X-ray simulation, validation of these components, results demonstrating the PTC task simulation and, finally, conclusions.

2 Related Work

2.1 Respiratory Motion

A range of techniques exist to improve image acquisition during respiration, including gating, real-time tracking, and magnetic tracking, for example when planning lung radiotherapy. The breathing cycle variations could even be predicted with spirometer, laser displacement sensors, markers on the body, etc. In IR though, compensating for organ motion when using real time imaging to direct a needle into a moving visceral target requires specific operator skills. Simulation using mathematical formulations is the focus of our approach to modeling the PTC task to train these skills.

Von Siebenthal *et al.* use previously acquired images to predict respiratory motion [43]. However, the breathing cycle is not reproducible and makes this approach unsatisfactory. This problem is overcome in [21] by using on-line ultrasound to register magnetic resonance imaging (MRI). It is not a parameterizable mathematical model and therefore it cannot be applied to a simulator.

Other solutions make use of physically-based models that can be tuned to match different breathing cycles. Two main approaches have emerged.

The first approach uses accurate mechanical models employed to target lung tumours in radiotherapy. Finite element methods are used to solve the continuous mechanics equations. Kyriakou *et al.* model the lung as a cylinder with various internal layers and the diaphragm acting as a piston [22]. In contrast, Didier *et al.* model the ribs with a helicoidal-based motion and the motion of the lungs is computed with contact detection and sliding [11]. The effect of contact surfaces and hyperelastic material properties on the mechanical behavior of human lungs has also been investigated [2]. Pato *et al.* studied the diaphragm with shell elements and the boundary condition is a uniform pressure applied in a radially on the muscle part of the diaphragm [30]. These models give reliable results, but they are far from interactive, mainly because of their non-linear nature.

Alternatively, the second approach makes use of heuristic models to achieve real-time performance. One of these models makes use of simple geometrical transformations only. For example, the use of a geometrical model based on a parametric surface has been proposed to build a trunk model [33]. The use of particle systems has also been investigated. A particle system in this context is a set of punctual masses moving under external as well as internal actions. The particle behavior in response to these forces is computed using physical laws. Implicit surfaces are often fitted to the shape of the object defined by the particles [23]. In [18] the author presents a method to compute lungs deformation with a particle method. The boundary conditions are given by diaphragm and ribcage actions. The diaphragm vector field is given by template matching method and the ribcage motion is given by a kinematic method. They are both monitored by a breathing curve given by an analytical model. However, the integration of realistic tissue properties into particle models is not trivial. Mass-spring modeling is the more frequently used heuristic method. The nodes are punctual masses and a cohesion force, commonly based on linear elasticity, is applied to each node edge. This method has been used in [48, 31] to model respiration.

2.2 X-ray Simulation

Different methods of simulating X-ray imaging techniques, based on particle physics, are now available. There are two main classes of X-ray simulation algorithms. The first approach is probabilistic and based on Monte Carlo trials [3], while the second is determinist or analytic, based on ray-tracing [14] (these include solving the Boltzmann transport equation [17]).

The Monte Carlo method consists of individually tracking each photon during its different interactions with matter at each step of the simulation. This method can produce very accurate images, but they are computationally expensive. Recent effort has been made to use the graphics processing unit (GPU) to tackle some bottlenecks of the Monte Carlo software available from European Organization for Nuclear Research (CERN) (GEometry ANd Tracking 4 (GEANT4) [1]) [19].

The ray-tracing principle has been adapted to the simulation of X-ray imaging to provide a fast alternative restricted to the computation of directly transmitted photons. Radiation attenuation is computed by considering the amount of penetration of a ray into the object. Freud *et al.* proposed a modified version of the Z-buffer, known as the *L*-buffer (for length buffer), to store the length of a ray crossing a given 3D object [14].

Existing GPU-accelerated simulations of X-ray images are mostly based on volume data [12]. More or less accuracy, depending on the end-user's application, can be implemented. For example, when only real-time visual feedback is acceptable, the length of a ray passing through a voxel can be an approximation [45]. In DRR computations, real-time performance does not need to be achieved as long as the results are numerically accurate and the rendering time remains clinically acceptable.

Little research has been reported on GPU-accelerated simulations of X-ray images from polygon data. This approach has been used in an interventional cardiology trainer [10], but no details about the implementation or the level of accuracy of the physics models used have been published.

Realistic central processing unit (CPU)-implementations that make use of triangle meshes do not achieve interactive frame rates. In [37], we demonstrated that this can be achieved using a GPU-implementation based on the *L*-buffer technique. In [14] and in [25], two algorithms were proposed to handle artefacts due to robustness issues in the *L*-buffer, but these methods are not efficiently applicable on the GPU. We proposed an alternative method using an adaptive filtering to solve this problem. However, the simulation was restricted to directly transmitted photons in the monochromatic case and only single point sources were taken into account. These approximations produce visually convincing images, but the method is not realistic enough for most physics-based applications. For example, the focal spot diameter of medical X-ray tubes ranges from 0.3 to 2.0mm [34]. In [38], we demonstrated that it is possible to take into account the shape of the focal spot of the X-ray tube, as well as polychromatic X-ray beam spectra.

3 Methods

The two core components of our enhanced simulator are: i) the breathing motion, and ii) the X-ray imaging. To take full advantage of the computational resource of the computer, the motion due to respiration is calculated on the CPU and the X-ray imaging is simulated on the GPU only. These respiration and fluoroscopy simulations have been integrated within a VE that includes a virtual patient containing the necessary anatomical models, which correspond to polygon meshes obtained after the segmentation of patient specific CT data. Meshes are only composed of surface triangles. They have been decimated and smoothed to offer a good compromise between computing time and visualisation on the fluoroscopy screen.

3.1 Respiratory Motion Modelling

The main objective is realism with interactive speed. We must cater for the target(s) of the needle puncture procedure, anatomy visible in the X-rays (e.g. lungs, pleural reflection), and the organs

producing the respiratory motion (e.g. the diaphragm, ribs). Our simulation must dynamically update the vertex positions of the meshes defining these organs based on physiological studies.

3.1.1 Physiology

Breathing is a complex process resulting from the action of several muscles. The primary muscles of respiration include the intercostal muscles and the diaphragm. A physiological study based on [28] was used to set the boundary conditions of our model. During inhalation, the contraction of both the intercostal muscles and the diaphragm lowers the air pressure in the lungs and causes air to move in. During exhalation, the intercostal muscles and the diaphragm relax. The two different actions from the intercostal muscles and diaphragm are considered separately and will be controlled by two independent parameters, which will then govern the natural deformation of the other organs (lungs, liver, *etc.*).

It is important to note that there are different kinds of breathing, for example calm breathing and the casual involuntary breath of healthy individuals that involves a considerable amount of diaphragm motion.

In the literature various models have been studied to reproduce the real human respiration time-function pattern [33, 27]. In our simulator, we use the simplest model (Eq. 1) as our main concern is the computing time. Similarly, in our validation section, we evaluate the best model parameters without time interpolation and therefore independently of the breathing curve analytic model. Eq. 1 gives a muscle action intensity $f(t)$ at a time t given its maximum *Amplitude* and at a certain *frequency*.

$$f(t) = Amplitude \sin(frequency \times t) \tag{1}$$

The method presented below can handle such variability, including hyperventilation or sudden breath holding.

3.1.2 Computational Requirements

Motion and soft tissue deformation must occur as fast as possible. We divided the organs into three categories with an increasing level of complexity: static organs, rigid organs that follow a kinematic motion and deformable organs. If we focus on the human trunk, only the spine can be assumed as static. The ribs and the sternum are rigid, but move following a kinematic law. All the other organs will be deformed. We have deployed the Chain Mail algorithm to model soft tissue as it is efficient and suitable for real time interaction [15]. When only using a few points, the Chain Mail algorithm performs better than any other methods based on time integration, which require the computation of forces, acceleration, velocity and position for each node, and for a time step small enough to allow convergence.

3.1.3 Chain Mail Algorithm

Our implementation is based on the 3D Chain Mail algorithm extension proposed by Li and Brodlie [26]. In this model, the object is defined as a set of point elements. The elements are interconnected as links in a chain, allowing each point to move freely without influencing its neighbours, within certain pre-specified limits. When an element of the object is moved and reaches this limit, the neighbours are forced to move in a chain reaction that is governed by the stiffness of the links in the mesh. Let α_{min}, α_{max} and β be the controlling parameters for compression, stretching and shearing respectively.

3.1.4 Intercostal Muscles Model

The rib movement produced by the intercostal muscle can be assumed to follow a kinematic motion [44] that is a combination of two movements: an increase of the lateral excursion of the ribs and an increase of the anteroposterior diameter of the thorax. These two rotations are

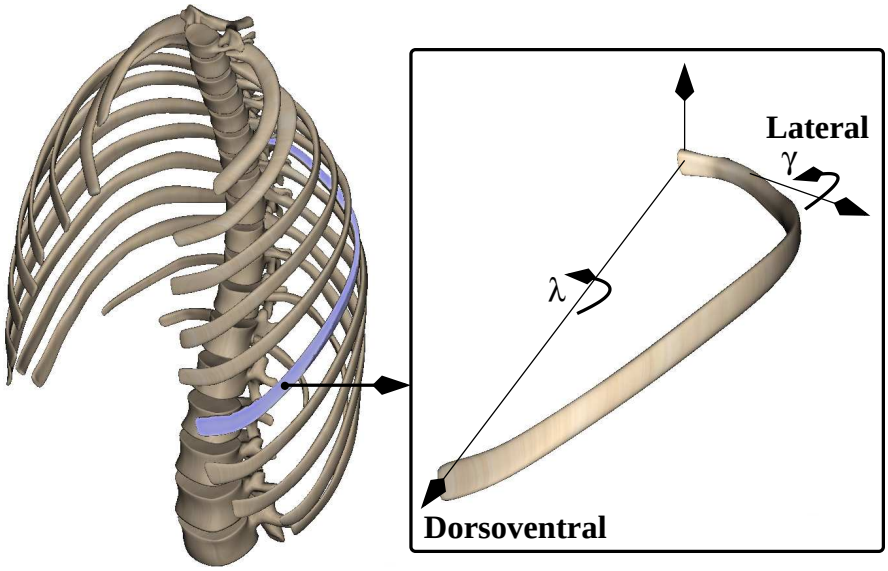


Figure 1: Ribs kinematics: the different rotation axis for the ribs.

respectively defined by γ and λ (see Fig. 1). They were measured for five subjects at functional residual capacity (FRC) and total lung capacity (TLC). In our model, the rotation angles are given by the following equations:

$$\gamma = ((\gamma_{\mathrm{TLC}} - \gamma_{\mathrm{FRC}}) \times \tfrac{1}{2} \times (1 + \sin(2\pi \mathit{frequency} \times t)) \quad (2)$$

$$\lambda = ((\lambda_{\mathrm{TLC}} - \lambda_{\mathrm{FRC}}) \times \tfrac{1}{2} \times (1 + \sin(2\pi \mathit{frequency} \times t)) \quad (3)$$

3.1.5 Diaphragm Model

During inspiration, the diaphragm muscles contract, the dome descending while maintaining a nearly parallel orientation to its original location. During expiration, the diaphragm relaxes, returning passively to its equilibrium state. However, it is not just the contraction of the muscles that governs the movement of the diaphragm (see Fig. 2) as it is attached to the spine and the lower ribs, and therefore follows the ribcage. In our simulation, the diaphragm mesh is composed of about 2000 elements, with almost half of them belonging to the central tendon, and the other half to the muscle part.

The algorithm also simulates ''hysteresis" effects, i.e. the motion paths simulated in inspiration and expiration are not the same. It is achieved by the Chain Mail algorithm: pulling mesh elements is different than pushing them. Two parameters are involved respectively stretching α_{max} and compression α_{min}. The displacement path of the diaphragm vertices varies when the diaphragm tendon is going upward or downward.

We model the tendon by setting α_{min} and α_{max} to 1 and β to 0. It corresponds to the values needed to have a uniform displacement. It means that, if a displacement of ΔA is applied on A (the vertex that is being moved), the displacement on its neighbour vertex B will always be of ΔA because the bounding box around B defining if displacement occurs will be of null size. We can then have the uniform translation expected for the rigid tendon. The nearly-vertical movement is performed by forcing one central point to have a sinusoidal movement along the vertical axis. As the links for the central tendon are rigid, the points corresponding to this tendon will follow this movement. The other points, considered to be muscle tissue, will have the expected "chain reaction" according to the Chain Mail rules. In a preliminary step we manually tune the parameters based on qualitative criteria. The chosen values are $\alpha_{min} = 0.7$, $\alpha_{max} = 1.1$ and $\beta = 0.1$ for the shear parameter. The set up was guided by trying to achieve the highest possible level of realism. Given the course of the diaphragm being between 0 and 2cm and its height being

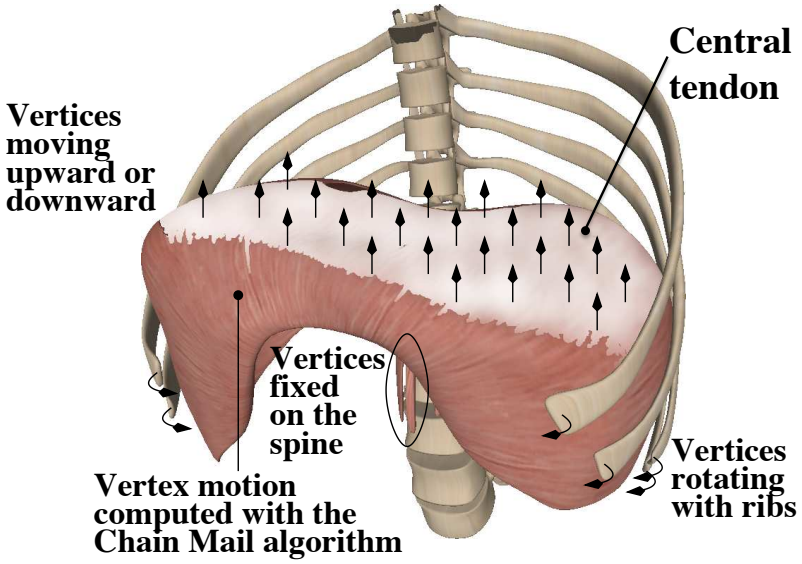


Figure 2: Diaphragm: the different vertex behaviors during exhalation.

around 10cm, the compression could be up to 20%. The model should then be able to contract smoothly and rapidly. This is achieved by setting the compression threshold α_{min} to only 70% of the distance between two nodes. However, the stretching parameter must be much higher to allow the muscle to quickly return to the relaxed position of the diaphragm (110% of the distance). The shear parameter has actually almost no effect in this case on the diaphragm behaviour. The muscle fibres are mainly arranged in the craniocaudal direction and the translation is only in this direction. There is then no shearing. To summarize, the behaviour obtained for the diaphragm is satisfactory in our simulator context. However in order to have more accuracy the parameters have been optimised in a second step (see Section 4.2 for the parameter optimisation).

As mentioned previously, the diaphragm must follow the ribs movement. This leads to the notion of 'linked organs' as discussed below.

3.1.6 Linked Organs

A valid approach to linking the organs would be to move the targeted organs by means of collision detection and collision response [47]. However, because of the high number of polygons present in the scene graph, this method is not feasible at interactive frame rates. Alternatively, areas common to two adjacent organs can be linked together. The vertices at the linked regions are then moved simultaneously. We choose to apply coupling forces instead of displacement constraints to have a continuous smooth deformation. This approach requires a pre-processing step to identify candidate linked regions whose distance must be below a given threshold. The threshold varies depending on the organs, e.g. very small for the diaphragm and liver, around the same value as the fat tissue thickness for the ribs and skin. A minimum distance to set rigid links ($\alpha_{min} = \alpha_{max} = 1$ and $\beta = 0$) is added to reduce the computing time. The organs linked together are:

- Diaphragm to the lower ribs,
- Lungs to the upper ribs,
- Lungs to the diaphragm,
- Liver to the diaphragm.

All calculations are performed using the CPU only. The GPU capability is then available to be used in parallel by the fluoroscopy simulation task.

3.2 X-ray Modelling

The aim of this fluoroscopy simulation is to produce a realistic X-ray image (restricted to directly transmitted photons) in realtime from the dynamic models described above. This can be calculated using the Beer-Lambert law and implemented on the GPU. Two implementations are available and they are compatible with a wide range of hardware/software plateforms. One supports OpenGL 2.x for older generations of graphics cards and renderers; one supports modern OpenGL 3.x/4.x implementations. No depreciated functions and no fixed-rendering pipeline functions are used. The library relies on OpenGL Shading Language (GLSL) and is portable: Intel, AMD, and Nvidia cards have been seamlessly used under Linux, Mac OS and Windows operating systems (OSs).

3.2.1 X-ray Attenuation Principles

The attenuation law, also called the Beer-Lambert law, relates the absorption of light to the properties of the material through which the light is travelling. The integrated form for a polychromatic incident X-ray beam (i.e. incident photons do not have the same energy) is:

$$\int E \times N_{out}(E)\mathrm{d}E = \int E \times N_{in}(E) \times \exp\left(-\int \mu(\rho(x), Z(x), E)\mathrm{d}x\right)\mathrm{d}E \tag{4}$$

with $N_{in}(E)$ is the number of incident photons at energy E, $N_{out}(E)$ is the number of directly transmitted photons and μ is the linear attenuation coefficient (in cm^{-1}). μ depends on: i) E - the energy of incident photons, ii) ρ - the material density of the object, and iii) Z - the atomic number (or, in case of a mixture, the chemical composition) of the object material.

Consider the geometry setup described in Fig. 3. This is a 2D representation of a scene made

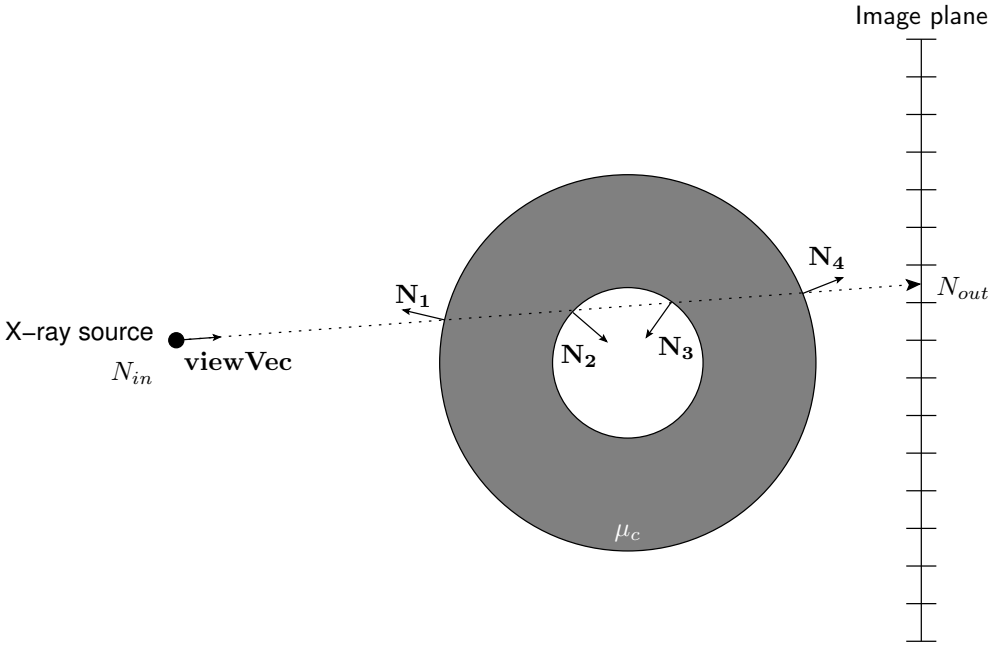


Figure 3: Principle of the computation of X-ray attenuation in the case of a simple geometric scene.

up of a circle in which a hole has been made (assuming no interaction in the medium outside the circle). Let E_{in} be the incident energy of a monochromatic X-ray beam (i.e. all the incident photons have the same energy). μ_c is the attenuation coefficient of the circle at this energy. The energy fluence (i.e. the amount of energy received by pixels of the detector) is computed as follows

using Equation 4:

$$E_{in} \times N_{out} = E_{in} \times N_{in} \times e^{-\mu_c([d_2-d_1]+[d_4-d_3])} \tag{5}$$

with d_i the distance in centimetres from the X-ray source to the respective intersection of the ray with an object. The parameters that need specifying are therefore: tissue geometry; attenuation properties of the tissue; the photon energy; and the position of the X-Ray source.

3.2.2 Tissue Properties

Only an organ surface needs modeling and standard triangle meshes can be used. Assuming normal vectors are outward, for each triangle mesh, the attenuation coefficient at a given energy level needs to be known.

The Hounsfield scale is used to specify the attenuation property of tissues in CT data sets. A value H in Hounsfield unit (HU) is given by:

$$H = 1000 \times \left(\frac{\mu}{\mu_w} - 1 \right) \tag{6}$$

with μ_w the linear attenuation coefficient of water at a given energy. Unlike attenuation coefficients, Hounsfield values do not depend on the energy of incident photons. Therefore, we use this standard scale to set the attenuation property of tissues within our simulator. Given Equation (6) and a database of attenuation coefficients for water at any energy from 1 keV to 100 GeV [5]it is theoretically possible to retrieve the attenuation coefficient from any Hounsfield unit at energies within that range. We demonstrate in Section 4.3.1 that this assumption is not correct and produces incorrect μ values.

To address this problem, we adopt the method proposed by Schneider *et al* to model tissue properties from HU values [32]. It supplies mathematical models to compute densities and atomic composition of human tissues from HU values. Atomic compositions are used in conjunction with a database of photon cross sections to compute mass attenuation coefficients. Mass attenuation coefficients can then be used with the densities to compute the linear attenuation coefficients requited to solve the Beer-Lambert law. In our implementation, we use the XCOM 3.1 database provided by National Institute of Standards and Technology (NIST) [5].

Note that the density of the lungs will change depending on the respiration step. The average HU value of the lungs was extracted for each volume of the 4D CT dataset. The HU value assigned to the lungs is iteratively modulated to take into account this change of density. For this purpose, a sine wave is used in a similar manner as Eq. 1.

3.2.3 Beam Spectrum

The photon spectral distribution of the incident beam has to be taken into account to produce physically realistic images. The photon spectrum can be divided into a set of discrete energies. To each energy value corresponds a photon number that is a fraction of the whole spectrum. So far, we have used the Birch and Marshall catalogue [6]. Fig. 4 shows the beam spectrum of an X-ray tube at 90 kV peak voltage whose output is filtered to remove low energy photons as it is the case in medical X-ray tubes. The mean energy is 80 keV, the energy typically used for the acquisition of fluoroscopic images of the abdomen.

3.2.4 Source Shape

When the size of the X-ray source is small enough, the blur due to the finite size of the source (usually called 'geometrical unsharpness') can be neglected. In this case, the X-ray emitter is modelled using a single 3D point source only. However, the geometrical unsharpness can significantly degrade the image spatial resolution and contrast and so is taken into account to perform a more realistic simulation. Depending on its finite size and shape, the source is then sampled into a set of elementary source points. A fraction of the number of incident photons is then equally distributed to all the source points.

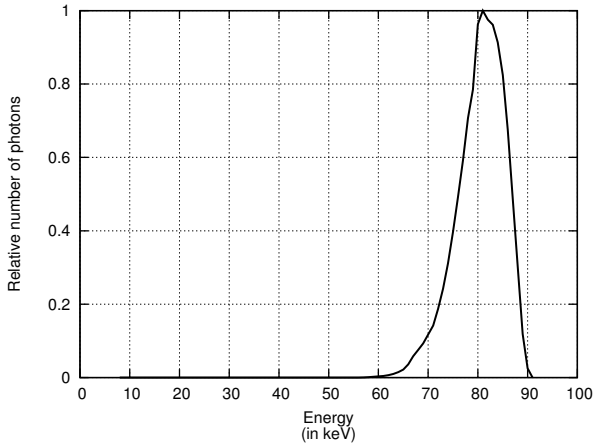


Figure 4: Spectrum of the incident polychromatic beam (90 kV X-ray tube peak voltage). The output of the X-ray tube is filtered using two copper plates (5 mm each).

3.2.5 The *L*-buffer Principle

To efficiently compute the path length (L_p) of a ray through an object, we use Freud *et al.*'s *L*-buffer algorithm [14]. For each scanned object an image (or *L*-buffer) is produced. The intensity of each pixel corresponds to the distance covered by the ray within the object, from the source to the pixel centre. Fig. 5 shows an example of *L*-buffer for the cube used in Fig. 13.

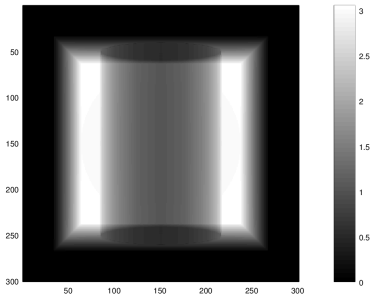


Figure 5: Example of *L*-buffer.

Fig. 3 shows that the ray penetrates into the object when the dot product between direction vector of the ray and the object surface normal is negative. This dot product is positive if the ray leaves an object. Thus the path length of the ray into an object can be written as follows:

$$L_p = \sum_i (\text{sgn}(\mathbf{viewVec} \cdot \mathbf{N_i}) \times d_i) \tag{7}$$

where **viewVec** is the *viewing vector* (or the direction vector of the ray) (i.e. the unit vector from the emission point to the detector's pixel); i refers to the i^{th} intersection between the ray and the object surface; d_i is the distance (in cm) from the X-ray source to the i^{th} intersection point; $\mathbf{N_i}$ is the vector normal to the object surface at the i^{th} intersection; $\text{sgn}(\mathbf{viewVec} \cdot \mathbf{N_i})$ is the sign of the dot product between **viewVec** and $\mathbf{N_i}$.

Note that intersections are found in an arbitrary order. One advantage of this approach is to avoid sorting through the intersections. It makes our implementation of *L*-buffer very effective.

3.2.6 The Simulation Pipeline

The hardware-accelerated implementation presented in [37] was restricted to directly transmitted photons in the monochromatic case and it did not take into account geometrical unsharpness. It was implemented in C++ using the OpenGL application programming interface (API) and GLSL. Here we extend the simulation pipeline using additional passes to take into account both the energy beam spectrum and the finite size of the X-ray source.

The principle of computing direct images is to cast rays from the X-ray source to every pixel of the detector. For each ray, the total path length through each object is computed using geometrical computations. The attenuation of X-rays for a given pixel is then computed using the recorded path lengths. Finally, the energy deposited by photons is integrated for every pixel of the detector. The algorithm is divided into successive building blocks. Each block corresponds to a 'rendering pass'. Only the final rendering pass is displayed on the screen. Intermediate rendering passes use frame buffer objectss (FBOs) and are stored into textures, these components are required for fast off-line rendering.

In the simple case when the beam spectrum is monochromatic, applying Equation (4), the energy received by the detector can be written as follows:

$$I_{out} = I_{in} \times \exp\left(- \sum_{i=0}^{i<objs} \mu(E,i) L_p(i) \right) \tag{8}$$

with E the photon energy, $objs$ the total number of objects, $L_p(i)$ the value of the L-buffer for Object i, I_{in} the input intensity that corresponds to $N_{in} \times E$, and I_{out} the output intensity. Details about the implementation strategy to compute Equations 7 and 8 can be found in [37].

The new X-ray attenuation pipeline treats skin as a special case. Indeed, its geometrical model is restricted to its external surface only. Therefore, the L-buffer of any internal structures must be subtracted from the skin surface L-buffer. The L-buffer of the skin is then given by:

$$L_p(skin) = L_p(skinSurface) - \sum_i L_p(i) \tag{9}$$

with $L_p(skinSurface)$ the L-buffer of the skin surface only. In this case, an extra FBO is needed to compute $\sum_i L_p(i)$. This is called $FBO\left(\sum_i L_p(i)\right)$. Note that the attenuation coefficient associated with the skin corresponds to fat.

An additional loop is added to take into account the beam spectrum. This is required to integrate the energy deposited by each energy channel into the final image.

Another loop is also added to take into account the geometrical unsharpness.

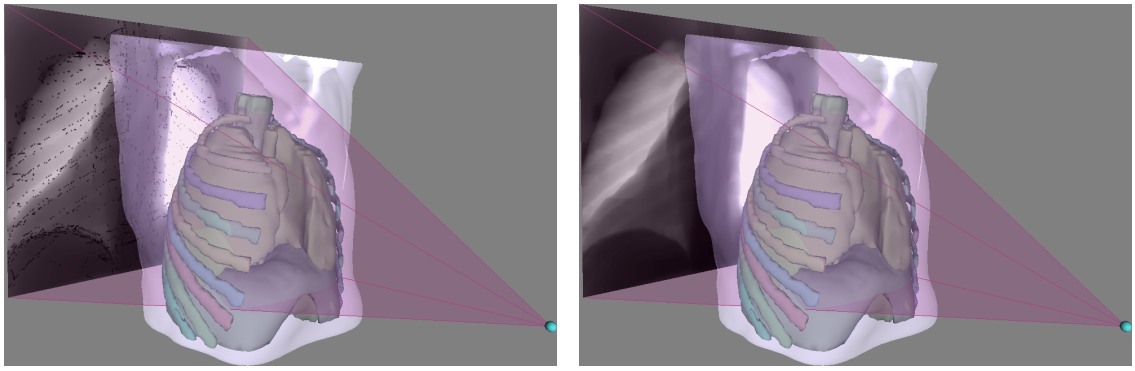
3.2.7 Adaptive Filtering for Artefacts Reduction

When intersections occur between a ray and an object, it is assumed that there are as many incoming and outgoing intersections. However, some intersections may be duplicated or missed when the ray hits an edge or a vertex of a triangle. When the normal vector $\mathbf{N_i}$ is perpendicular to the viewing direction **viewVec**, artefacts can also occur. In this case, depending on the normal of the triangles, the computed thickness will be either very high or negative. This will lead to black or white pixel artefacts in the final image (see Fig. 6(a)).

These issues are addressed in [14] and [25] in the case of a CPU implementation, but these solutions are not easily portable in the case of GPU programming. However, it is possible to detect on the GPU for each pixel if such artefacts will occur. Indeed, Equation (10) should always be null for every pixel:

$$\sum_{i=1}^{n} \mathrm{sgn}(\mathbf{viewVec}.\mathbf{N_i}) \tag{10}$$

with n the number of intersections between the ray and the processed triangle mesh. The fragment shader used to compute the L-buffer is then extended to store the sign of the dot product into the



(a) Without filtering. (b) With adaptative filtering.

Figure 6: Effect of the artefact correction filtering.

green channel of the L-buffer texture. The sum operation in Equation (10) is performed taking advantage of the blending function used during the L-buffer computations.

Finally, before fetching any red component of the L-buffer texels, we check the validity of their respective green component. If the green component is not null, then the red value is invalid. To avoid the artefact, it is replaced by the average value of the valid texels within its direct neighbourhood. If no valid texels are found, we increase the size of the neighbourhood. Fig. 6(b) shows the X-ray image corresponding to Fig. 6(a) when artefact correction is enabled.

4 Results

The validation of medical simulators is typically performed by medical subject matter experts who assess the virtual environment using questionnaires and recorded metrics from the simulator. For an objective assessment, we have performed a quantitative validation of the respiration and of the X-ray simulations to actually measure errors in the simulation. This has been achieved by quantifying the difference between results of the simulation and reference data.

4.1 Patient-based Simulation

To validate our model of respiratory motion, we apply our method to meshes extracted from specific patient data sets. We perform the experiments on five patients and we validate the displacements of diaphragms and livers. The datasets include two 4D CT scans with 10 time steps in the breathing cycle. The three others are 3D CT scans with breath hold at inhale and exhale states.

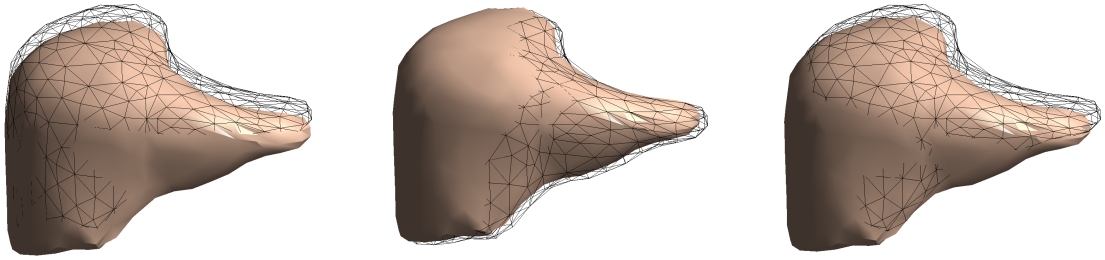
From these medical images, the organs required by the simulator were segmented using ITK-SNAP [46]. Lungs were automatically extracted while the diaphragm and the liver were manually segmented by a medical expert for ten steps in the breathing cycle of the two 4D CT scans, plus the two breathing states (inhale and exhale) of the 3D CT scans. Ribs and spines are also manually segmented in order to properly separate the different ribs. The rotation centers are computed using the inertia matrix as described in [42].

Tab. 1 shows CT scan informations for each patient studied. Particularly resolutions and time step numbers are indicated.

This gave a total of 52 manually segmented organs. On average, the segmentation stage takes two hours per liver and three hours per diaphragm. A marching cube algorithm is then applied to extract a mesh from the isosurface. The resulting triangulation is smoothed and decimated into a mesh of about 2000 vertices with Voronoi Parallel Linear Enumeration (Vorpaline) [24].

Table 1: CT scan specifications for each patient studied

	Patient 1	Patient 2	Patient 3	Patient 4	Patient 5
Dimension	512 × 512 × 136	512×512×75	512 × 512 × 141	512 × 512 × 139	512 × 512 × 287
Voxel size $[mm^3]$	1.08 × 1.08 × 2.5	0.98×0.98×5	0.977 × 0.977 × 2	1.17×1.17×2	0.709 × 0.709 × 1
Respiration stage	2	2	10	2	10



(a) Diaphragm influence. (b) Ribs influence. (c) Combined influence.

Figure 7: Influence of the customised breathing on liver deformation and motion. Mesh in black indicates original position of the liver.

The rib rotation parameters and the tendon force were tuned to match the patient's breathing cycle as much as possible. It was observed that the breathing was mainly diaphragmatic and not thoracic. No data with significant thoracic breathing was available. This is because the datasets were acquired during a radiotherapy treatment planning in which patients were lying on their back and asked to breath quietly. A more random behaviour could occur during abdominal IR procedures. To address this, we reproduced different kinds of breathing on a virtual patient and studied the influences on the liver of three kinds of breathing: only diaphragmatic, only thoracic and both influences combined. In Fig. 7(a), we can see that the liver had mainly an axial translation due to the diaphragm contraction. In Fig. 7(b), we can see that the ribs pulled the liver surface in contact while the ribcage expands. In Fig. 7(c), we can see the combined influence of respiration muscles applying both an axial translation and a small lateral movement.

4.2 Respiration Parameter Optimisation

Manually tuned parameters were initially utilized in our application. Tuning the parameters of such simulators is still often performed by hand using trial and error. This is time consuming and prone to error when the number of variables increases and it can lead to large numerical errors. We showed that a much better methodology can be easily deployed using an evolutionary optimization scheme [39].

The patient data can only be used to validate our diaphragm model and its influence on the liver. The degrees of freedom of our models are 15 values included within four parameter sets defined as follows:

1. The chain mail parameters for the liver and the diaphragm, namely the compression α_{min}, the stretching α_{max} and the shearing β. They are three mechanical parameters that are different for each patient.

2. The plane separating the muscle part from the tendinous part of the diaphragm as explained in Section 3.1.5. We characterise it by the equation (11) defined by four parameters a, b, c and

d linked to the anatomy of the patient and related to the vertices $\mathbf{P_{tend}}(x_{tend}, y_{tend}, z_{tend})$.

$$a.x_{tend} + b.y_{tend} + c.z_{tend} + d < 0 \tag{11}$$

3. The distance defining the mechanical links as described in Section 3.1.6. We define one value for the distance between the diaphragm and the ribs, and one distance between the liver and the diaphragm. They are also parameters linked to the anatomy of the patient.

4. The final tunable parameter is the amplitude of the tendinous part of the diaphragm. It is a 3D vector $\mathbf{F_{tend}}$ similarly applied to all points $\mathbf{P_{tend}}$ such that their new positions $\mathbf{P'_{tend}}$ are defined by equation (12).

$$\mathbf{P'_{tend}} = \mathbf{P_{tend}} + \mathbf{F_{tend}} \tag{12}$$

It is possible to estimate the error between the simulated data and the real data segmented from CT scans. This error should be as low as possible. Our evolutionary algorithm automatically tunes the models and minimises the discrepancies between geometries S extracted from segmented CT and from simulated geometries S' [39]. The surface difference is estimated with three indicators according to [4]: the mean distance $d_m(S, S')$, the root mean square $d_{rmse}(S, S')$ and the maximum $d(S, S')$ distance, which is also the non-symmetrical Hausdorff distance.

To objectively quantify the experiments, we estimate the difference between S and S' at the same time step. We also compute the respiratory motion amplitude on the real data with the same measurement technique by evaluating the difference between initial geometries (full full inhalation S_I) and geometries S_t at a given breathing stage t. Two type of results were then obtained: results for the two extreme steps of full inhalation and full exhalation, and results for the whole breathing cycle. For each case, we compute:

1. The respiratory motion amplitude with values $d_m(S_I, S_t)$, $d_{rmse}(S_I, S_t)$ and $d(S_I, S_t)$.

2. The error with manually tuned parameters with values $d_m(S, S')$, $d_{rmse}(S, S')$ and $d(S, S')$.

3. The error with our optimisation method with values $d_m(S, S')$, $d_{rmse}(S, S')$ and $d(S, S')$.

Results are presented on the Figures 8 and 9.

Fig. 10 presents $d_m(S_I, S_t)$ on each vertex of the organs with a colour code proportional to its value. Whilst errors resulting from manually tuned parameters are already acceptable to reasonably predict the patient's breathing with our simulator application, it is possible to significantly improve the quality of the results with our optimisation method (e.g. the error reduction is up to 4.23mm for Patient 5's liver), even if errors remain non-negligible due to segmentation issues (e.g. errors in segmenting the diaphragm). The algorithm is user-friendly and fully automatic: the parameters are directly computed from two geometrical states. We can conclude that our model is able to reasonably predict the patient's breathing for our application.

As a qualitative validation, Fig. 11 shows the difference between the initial and final states of the simulation to evaluate the boundary conditions. It illustrates the negligible rib kinematics influence that is pulling the diaphragm laterally. The tendon action resulting in translation motion is also clearly visible. The Chain Mail elasticity ensures a smooth continuity in spite of these deformations. A junior doctor has confirmed that these types of motions are realistic notably saying: "The Chain Mail does give a sense that the diaphragm is a muscular flexible organ under tension".

4.3 X-ray

The aim of this section is to demonstrate the accuracy of our GPU implementation. We will investigate: tissue properties, monochromatism, polichromatism, Radon transform, CT reconstruction, shape and position of the X-ray source. Experimental results computed on GPU are either compared with theoretical values or with ground truth images simulated using state-of-the-art Monte

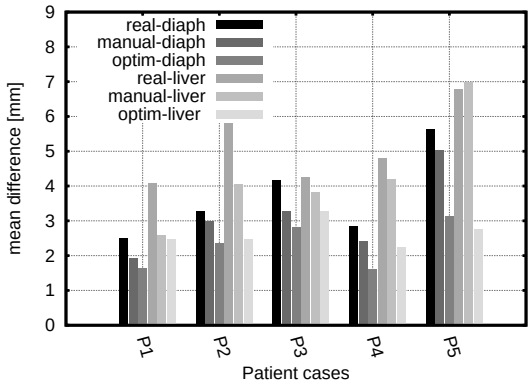
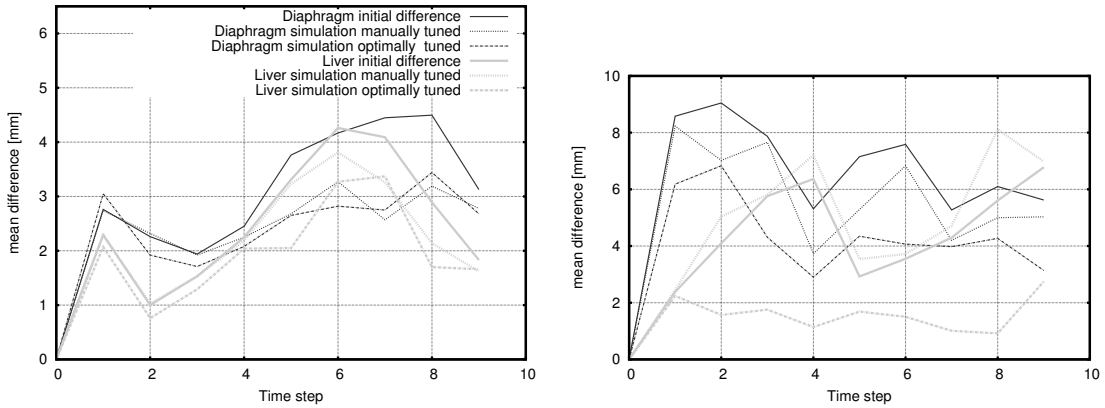


Figure 8: Amplitude, error on manual method and on optimisation method on the five patients from inhale to exhale.



(a) Patient 3.

(b) Patient 5.

Figure 9: Amplitude, error on manual method and on optimisation method on the ten time steps.

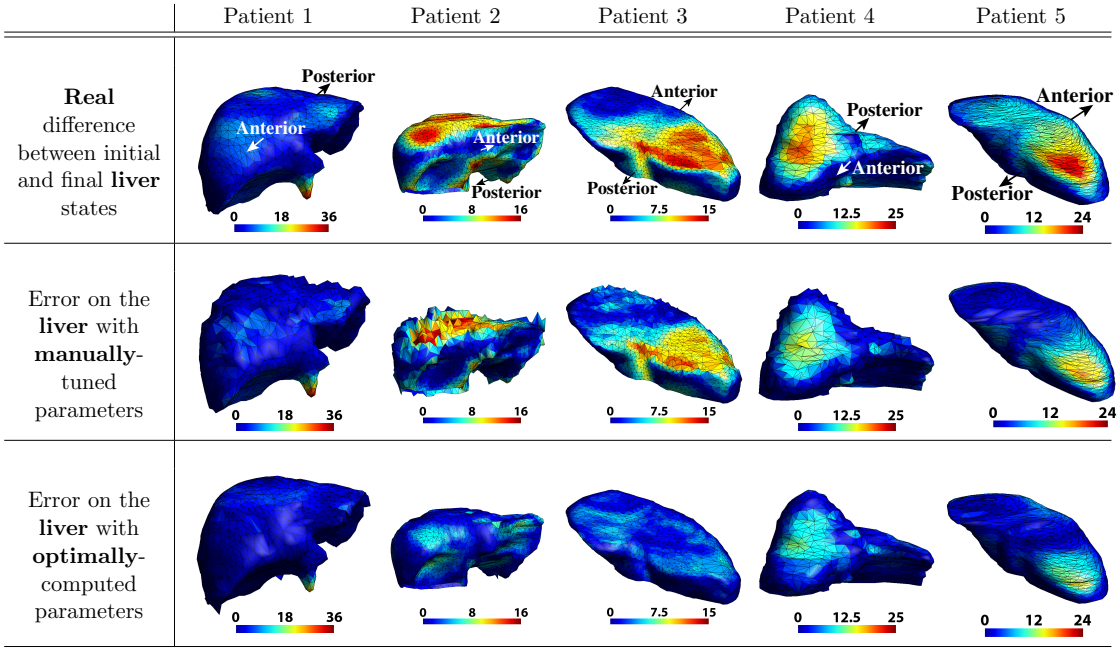


Figure 10: Amplitude, error on manual method and on optimisation method on livers of the five patients.

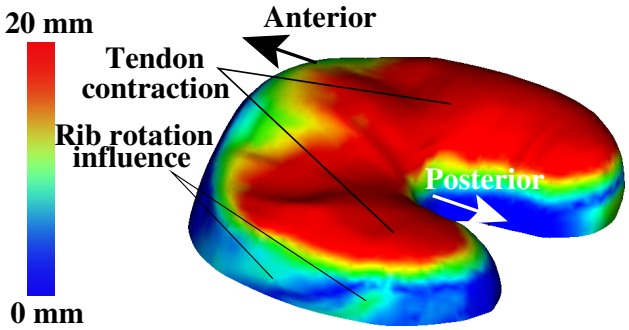


Figure 11: Qualitative validations of the diaphragm: difference between the beginning and the end of the simulation.

Table 2: Hounsfield unit for various types of tissues estimated at 80 keV using [16].

Material	ρ **(in g/cm^3)**	μ/ρ **(in cm^2/g)**	μ **(in cm^{-1})**	**HU**
Bone, Cortical (ICRU-44)	1.920E+00	2.229E-01	4.280E-01	1330
Tissue-Equivalent Gas, Propane Based	1.826E-03	1.783E-01	3.256E-04	-998
Tissue, Soft (ICRU-44)	1.060E+00	1.823E-01	1.932E-01	52
Water, Liquid	1.000E+00	1.837E-01	1.837E-01	0

Carlo (MC) software. The source code of all the validation tests, as well as the produced data, MC simulation scripts, etc. are available online at `http://gvirtualxray.sourceforge.net/validation/`.

4.3.1 Tissue Properties

Using Eq. 6, it is possible to compute HU values using the linear attenuation of water (μ_{water}) and tissues (μ) at a given energy. The NIST provides X-Ray mass attenuation coefficient (μ/ρ) and density (ρ) values for some human tissues [16]. μ can be retrieved using μ/ρ and ρ values. Using μ and μ_{water}, HU values can be estimated for any tissue as long as its linear attenuation coefficient is known. Tab. 2 provides a summary of values for various types of human tissues at 80 keV.

The naive approach to estimate μ for any tissue at a given energy is to use Eq. 6 again:

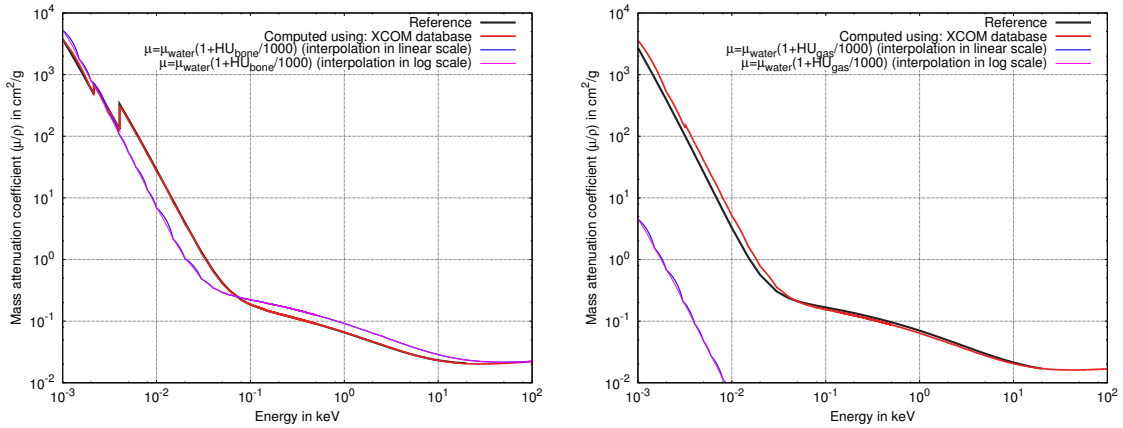
$$\mu = \mu_{water} \times \left(1 + \frac{H}{1000}\right) \tag{13}$$

It assumes μ_{water} at the corresponding energy and the Hounsfield unit of the tissue are provided. HU values are provided by reconstructed CT slices. Reference tabulated values of mass attenuation coefficients are provided by [16]. They are used to get the linear attenuation of water for energies between and 10^{-3} and 20 MeV (this range of energy is sufficient for most medical simulations). Note that such values are also used for comparison purposes in Fig. 12 as they provided reference data for various types of human tissues. As the mass attenuation coefficients of water are provided in a tabulated format, it is necessary to interpolate both the energy and mass attenuation data. Our implementation of Eq. 13 supports interpolation using a bilinear scale or a bilogarithmic scale. It can be observed in Fig. 12 that the bilinear scale in interpolation (blue curves) does not provide smooth results compared to the interpolation using bilogarithmic scale (magenta curves). Also, using Eq. 13 to compute attenuation coefficients of human tissues is not acceptable for realistic simulations. Estimated coefficients for bones and gas are significantly different from the reference data. This is because Eqs. 6 and 13 show linear relationships at a given energy between H, μ and μ_{water} whereas there is not such a linear relationship between the values of μ and μ_{water} at different energies.

This problem is tackled in our implementation using the method proposed by Schneider *et al* to model tissue properties from HU values [32]. Their model provides ways to estimate densities and atomic composition of human tissues from HU values. Using the atomic compositions and a database of photon cross sections, it is possible to compute mass attenuation coefficients. Fig. 12 shows that the values estimated with this method are closer to the reference data. The discrepancies can be explained by the differences in densities and atomic compositions between [16] and [32].

4.3.2 Monochromatism Case of Beer-Lambert Law

Now realistic input tissue properties are modelled, it is possible to compute X-ray attenuation. GPUs support two kinds of floating point numbers: 32-bit floats often called "single-precision"



(a) Bone, Cortical (ICRU-44)

(b) Gas, Propane Based

Figure 12: Comparison of three methods to compute mass attenuation coefficients (μ/ρ).

floats, and 16-bit floats often called "half-precision" floats. Our implementation supports both type of floats. Using 32 bits, we expect computations to be slower but more accurate than using 16 bits.

To assess the accuracy of our implementation, we consider a cube that has edge length of 3 cm. Its HU is 52 (soft tissue). A cylinder is inserted in the centre of the cube. The cylinder is made of bone (HU = 1330), its height is 3 cm and its diameter is 2 cm. We consider the energy of the incident beam is $I_0 = 80$ keV. From now on, we will use [32] and [5] to obtain linear attenuation coefficients (see Tab. 3). The energy orthogonally transmitted through the centre of the test objects is:

$$\begin{aligned} I_{expected} &= I_0 \times \exp\left(-\sum_i (\mu_i \times x_i)\right) \\ &= 80.000 \times \exp\left(-\left(0.3971 \times 2 + 0.1937 \times 1\right)\right) \\ &= 29.789 \text{ keV} \end{aligned} \tag{14}$$

Using half-precision floats on GPU, it is 29.831 keV; using full-precision floats, it is 29.789 keV.

Table 3: Tissue properties of bones and soft tissues estimated at 80 keV using [32] and [5].

Material	**HU**	ρ **(in g/cm^3)**	μ/ρ **(in cm^2/g)**	μ **(in cm^{-1})**
Bone	1329.886	1.804E+00	2.201E-01	3.971E-01
Soft tissue	51.715	1.063E+00	1.823E-01	1.937E-01

In the first case the relative error is 0.141%; in the second one it is 0.00%. These results are extremely close to the value that was expected.

4.3.3 Polychromatism Case of Beer-Lambert Law

Our implementation also supports the Beer-Lambert law in the polychromatic case, i.e. when there are photons of different energies in the incident beam. We consider the same test case as previously, but the incident beam is made of 10 photons of 100 keV, 20 photons of 200 keV, and 10 photons of 300 keV. Linear attenuation coefficients are shown in Tab. 4. The energy orthogonally

transmitted through the centre of the test objects is:

$$\begin{aligned} I_{expected} &= \sum_j \left(I_0(j) \times \exp\left(-\sum_i (\mu_{i,j} \times x_i) \right) \right) \\ &= (100 \times 10) \times \exp\left(-(0.3328 \times 2 + 0.1800 \times 1)\right) + \\ &\quad (200 \times 20) \times \exp\left(-(0.2369 \times 2 + 0.1445 \times 1)\right) + \\ &\quad (300 \times 10) \times \exp\left(-(0.2018 \times 2 + 0.1251 \times 1)\right) + \\ &= 4,353.175 \text{ keV} \end{aligned} \tag{15}$$

Using half-precision floats on GPU, it is 4,355.469 keV; using full-precision floats, it is 4,353.175 keV. In the first case the relative error is 0.053%; in the second one it is 0.00%. Once again, these results are extremely close to the value that was expected.

Table 4: Tissue properties of bones and soft tissues estimated at 100, 200, and 300 keV using [32] and [5].

Material	**HU**	**Energy (in keV)**	μ/ρ **(in cm^2/g)**	μ **(in cm^{-1})**
Bone	1329.886	100	1.845E-01	3.328E-01
Soft tissue	51.715	100	1.694E-01	1.800E-01
Bone	1329.886	200	1.313E-01	2.369E-01
Soft tissue	51.715	200	1.360E-01	1.445E-01
Bone	1329.886	300	1.119E-01	2.018E-01
Soft tissue	51.715	300	1.177E-01	1.251E-01

4.3.4 GPU vs MC: point source

Until now, a parallel beam of X-rays was used. Our implementation also support point sources, such as X-ray tubes. Using the same test object, we simulate two X-ray images:

- Using a MC method for particle physics implemented in GATE
- Using our GPU implementation

GATE is an opensource software developed by an international collaboration. Its focus is on MC simulation for medical imaging and radiotherapy. To achieve this GATE makes use of Geant4. It is CERN's MC simulation platform dedicated to particle physics in nuclear research (CERN is the European Organization for Nuclear Research).

Fig. 13 shows the geometry that we consider:

- The detector is made of 301 × 301 pixels. The size of each pixel is 0.3×0.3 mm^2.
- The centre of the detector is located at the coordinates 10 0 0 cm.
- The point source is located at the coordinates -10 0 0 cm.

Fig. 14 shows the images simulated using GATE (13.8 days of computations) and using gVirtualXRay. The two images are visually very close. The normalised cross-correlation between the images of Fig. 14 is 0.99747: The result of our GPU implementation matches the outcome of a MC simulation from a widely accepted tool.

4.3.5 GPU vs MC: Uncentered source

The source does not have to be centred on the detector. To test this, we move the position of the point source at the coordinates -15.0 0.5 0.5 cm. Fig. 15 shows the corresponding simulated images using GATE (12.9 days of computations) and our GPU implementation. Once again, the result of our GPU implementation matches the outcome of GATE. The normalised cross-correlation between the images of Fig. 15 is 0.99656.

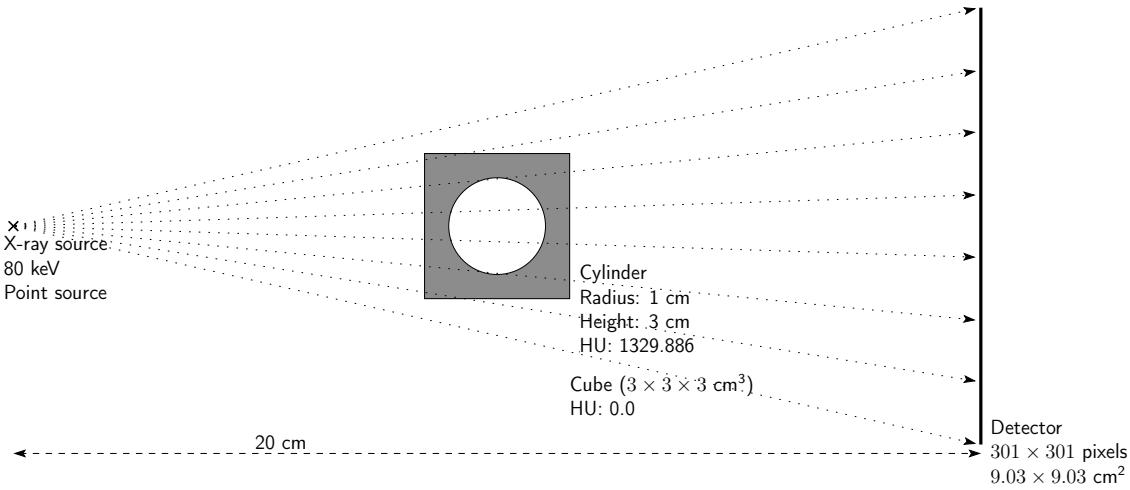
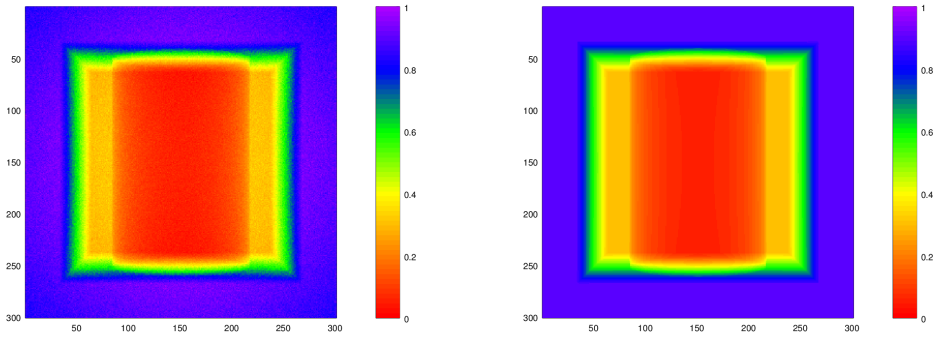


Figure 13: Simulation setup.



(a) Projection simulated with GATE.

(b) Projection simulated on the GPU.

Figure 14: Projection using a point source.

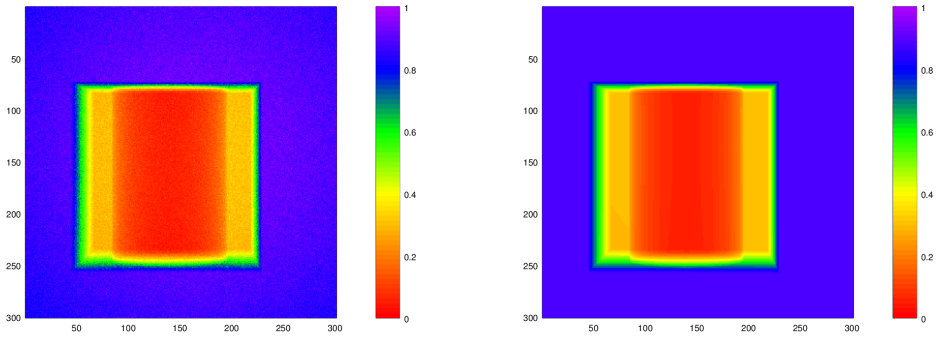
4.3.6 GPU vs MC: Cubic source

Our implementation also support *geometrical unsharpness*, e.g. when the source corresponds to a cube. To validate this functionality, we use a source corresponding to a cube, which has edge length of 0.5 cm. Its centre is located at the coordinates -10 0 0 cm. 14.4 days were required to obtain Fig. 16(a) with GATE. Fig. 16 is blurred compared to Fig. 14 (when a point source was used). This is the geometrical unsharpness. The two simulations seem to be extremely closed to each other. Normalised cross-correlation (NCC) (= 0.99743) demonstrates the validity of our approach.

4.4 Computational Performance

To assess the speed of computations, 500 iterations are computed with and without artefact correction on several computers. The tests are repeated 15 times. Computers bought in 2010, 2011, 2012, 2013 and 2014 have been used. The computing time is recorded. Tab. 5 shows the average number of X-ray projections and respiration deformations computed per second is recorded when 41,710 triangles are used in total. It shows that real-time performance is achieved on every tested platform, including 5 year old laptops.

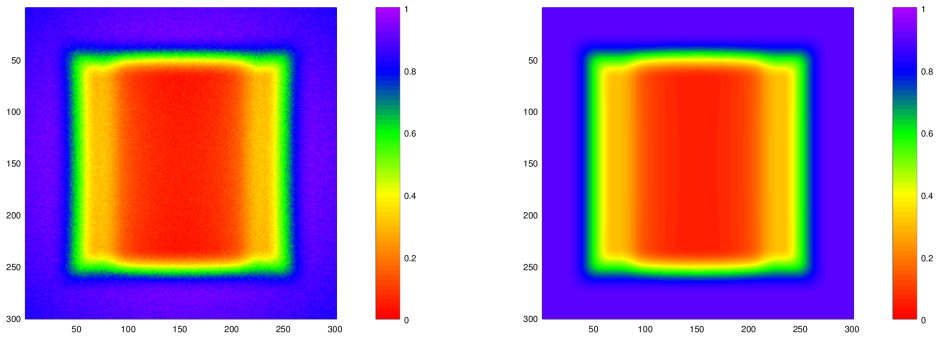
Figures 17 and 18 summarise the performance achieved with our latest PC for the respiration



(a) Projection simulated with GATE.

(b) Projection simulated on the GPU.

Figure 15: Projection using a source that is not centred.



(a) Projection simulated with GATE.

(b) Projection simulated on the GPU.

Figure 16: Projection using a cubic source.

and X-ray simulations respectively. It runs on AMD hardware. The data for the 15 runs is presented as boxplots. The horizontal axis corresponds to the total number of triangles used in the simulation. It is presented with a log-scale. The vertical axis corresponds to the number of iterations achieved in a second. Fig. 17 shows that the lower the number of triangles is, the faster the respiration simulation is. This is intuitive as the complexity of the algorithms are strongly related to the number of triangles. Both half and full floating point precisions are taken into account in Fig. 18. In [37], we saw that the performance using half precision was significantly faster than using full precision. Back then, support of full precision in programmable shaders was at an early stage. On today's generation of hardware, the difference of performance between half and full precisions is relatively small. When the number of triangles increases, the performance without artefact correction decreases. The bottleneck in this case is linked to the number of triangles. With artefact correction however, the performance does not decrease as much. This is because the bottleneck in this case is a fragments program, i.e. the number of pixels in the simulated image is the limiting factor. We have to keep in mind that there are less artecfacts to correct when the number of triangles is higher. In other words, the fragment program for artefact correction will be less solicited with a high number of triangles. As it makes use of `if` statements, branching occurs, which is a limiting factor of the single-instruction multiple-data (SIMD) architecture used in GPUs.

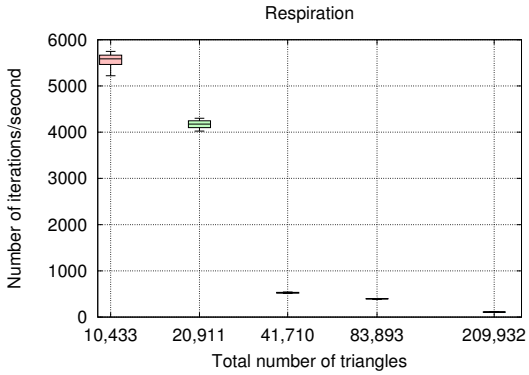


Figure 17: Computing performance of an AMD FX-8 8350 CPU for the respiration.

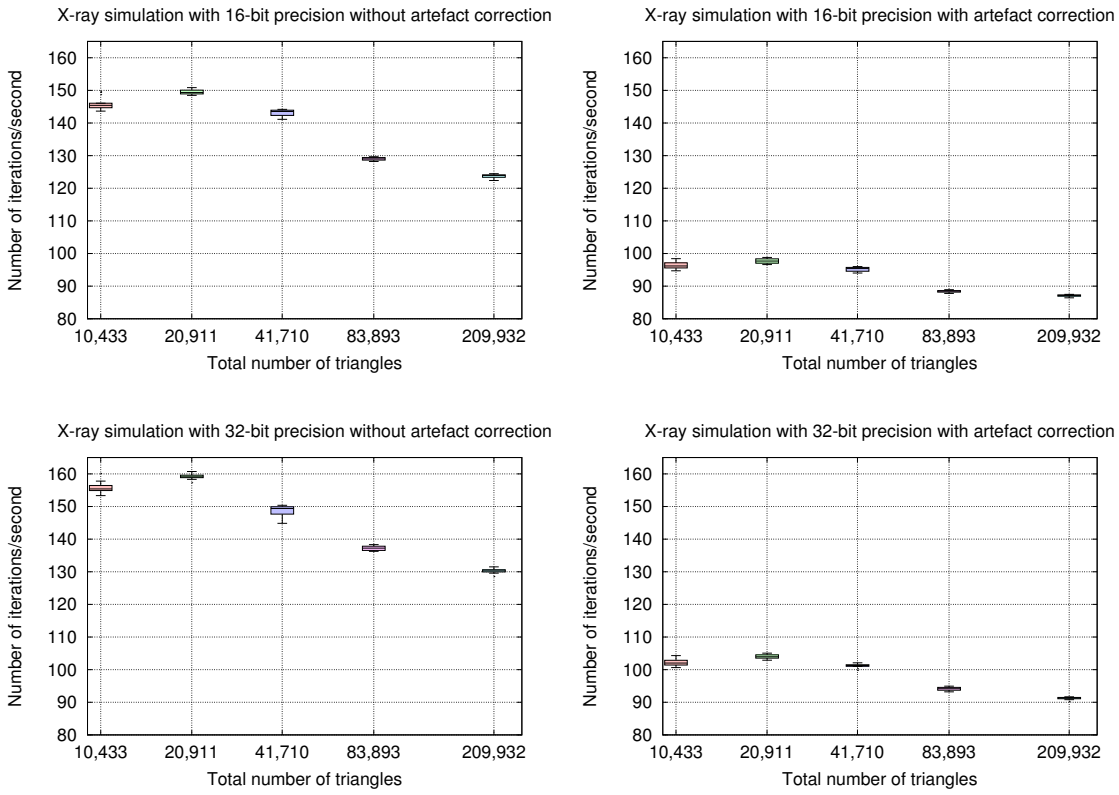


Figure 18: Computing performance of an AMD Radeon R9 285 GPU for the X-ray simulation.

Table 5: Performance comparison of different platforms. Results are given in frames per second (FPS). X-ray computations are performed using a point source, with full-precision floating point numbers, with artefact correction (1), without artefact correction (2).

CPU	GPU	(1)	(2)	Respiration	Year
AMD FX-8 8350 @ 4.00GHz	AMD Radeon R9 285	101	149	528	2014
Intel(R) Core(TM) i7-3770 @ 3.40GHz	NVIDIA Quadro K5000	77	115	612	2013
Intel(R) Core(TM) i7-3667U @ 2.00GHz	Intel HD Graphics 4000	13	49	315	2012
Intel(R) Core(TM) i7 X 990 @ 3.47GHz	NVIDIA GeForce GTX 580	91	92	718	2011
Intel Core 2 Duo @ 2.66GHz	NVIDIA GeForce 320M	16	31	221	2010

4.5 Combined validation of both components

Quantitative validation combining both components is extremely hard to conduct. For health and safety reasons, it is impossible to obtain both 4D CT and fluoroscopy videos of the same patient/volunteer. X-ray is a ionising radiation. Ethics forbids irradiating people without medical reasons. The radiation dose must be minimal to reduce the risk of cancer from medical imaging. X-ray imaging is a real-time modality. Typical fluoroscopy systems generate 30 frames per second. The typical respiratory rate for a healthy adult at rest is 0.20 to 0.33 breaths per seconds, which is about 100 times smaller than the framerate of a fluoroscopy system. The effect of movement in the X-ray image is then small enough to be ignored in real-time imaging. For these reasons, we have favoured to quantitatively validate the different components separately.

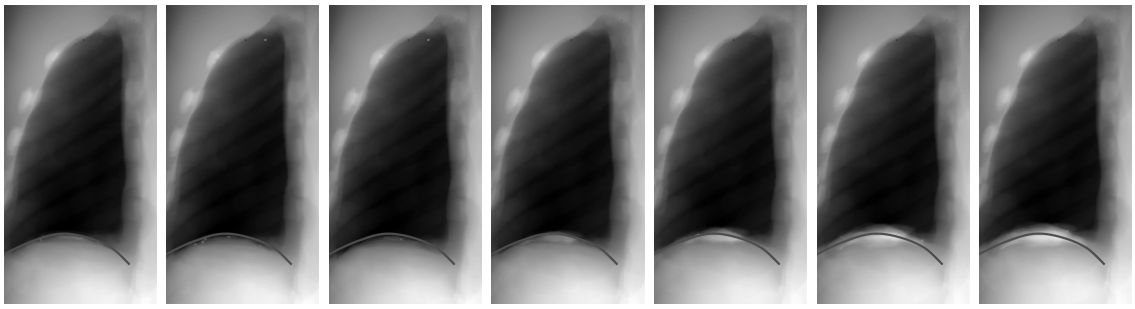


Figure 19: X-ray sequence focusing on the right lung, the grey mark indicates the position of ribs and diaphragm at the beginning of the breathing cycle.

It is however possible to describe the overall result qualitatively. Fig. 19 shows a respiration cycle focusing on the right lung. This sequence is extracted from the results of our optimization algorithm on Patient 2's data (see Section 4.2). It is mainly a diaphragmatic respiration pattern due to the acquisition procedure (radiotherapy) where patients are lying on their back. It is the reason why the diaphragm is having a bigger influence on the respiratory motion than the intercostal muscles. The diaphragm contraction and relaxation can indeed be noticed by comparing the white-black interface corresponding to the diaphragm with its initial position represented by the grey curve on Fig. 19. There is no artefact that could be due to mesh triangle becoming to small and normal inversion. The simulation is stable with time and no mesh explosion occurs.

5 Conclusion

In this paper, we have presented a real-time simulator whose architecture has been specifically developed to allow realistic multi-organ physically-based deformations with on-line fluoroscopy. Care has been taken to optimise the trade-off between the realism of the results and the speed of

the computations. We have developed an efficient technique for applying motion due to respiration to the virtual patient. The deformation module has been implemented on the CPU. After studying the physiology involved, we chose to control the respiration by the rib cage and the diaphragm. Ribs were modelled as rigid bodies with kinematic laws, while diaphragm motion was simulated by the up and down motion of the central tendon coupled with its rib attachments. Soft-tissue deformation was handled by an extended Chain Mail algorithm allowing fast multi-organ interaction. Our qualitative and quantitative validation study shows the effectiveness of the method. The X-ray simulation has been implemented on the GPU. It is a multi-pass algorithm using an OpenGL pipeline. For each X-ray pixel, the first pass computes tissue penetration (the skin is treated as a special case), the second computes an intermediary result required in the final pass to compute the cumulative attenuation using the Beer-Lambert law. To improve realism, two additional loops have been added to enable geometric unsharpness and polychromatism. This implementation is both fast and accurate.

We conclude that our hypothesis that these computationally intensive processes can run simultaneously and be integrated within an interactive VE running on a standard personal computer (PC) platform (inc. low tech laptops such as Macbook Air) has been proved to be true. A further future application might be the tracking and interactive visualisation of the movement of tumours in radiotherapy, where unrealistic computational times rule out using Monte Carlo simulations. This could be achieved by generating synthetic lung neoplasm data in real time, using patient specific models, segmented from CT data.

We also plan to improve organ deformation accuracy. We are currently investigating two avenues of research regarding this point: i) using finite-element modelling instead of the Chainmail method and ii) adding more degrees of freedom on the diaphragm motion model.

Acknowledgements

This work has been partially funded by FP7-PEOPLE-2012-CIG project Fly4PET – Fly Algorithm in PET Reconstruction for Radiotherapy Treatment Planning. We thank the Marie Currie Institute and the Centre Léon Bérard for providing the medical data sets that were used in the validation study.

Acronyms

API application programming interface.

CBCT cone-beam computed tomography.

CERN European Organization for Nuclear Research.

CPU central processing unit.

CT computed tomography.

DRR digitally reconstructed radiograph.

FBO frame buffer objects.

FPS frames per second.

FRC functional residual capacity.

GEANT4 GEometry ANd Tracking 4.

GLSL OpenGL Shading Language.

GPU graphics processing unit.

HU Hounsfield unit.

IR interventional radiology.

MC Monte Carlo.

MRI magnetic resonance imaging.

NCC normalised cross-correlation.

NIST National Institute of Standards and Technology.

OS operating system.

PC personal computer.

PTC percutaneous transhepatic cholangiography.

SIMD single-instruction multiple-data.

TLC total lung capacity.

VE virtual environment.

Vorpaline Voronoi Parallel Linear Enumeration.

VPH virtual physiological human.

VR virtual reality.

References

[1] S. Agostinelli and *et al.* GEANT4 - a simulation toolkit. *Nucl. Instrum. Methods Phys. Res., Sect. A*, 506(3):250–303, 2003. doi:10.1016/S0168-9002(03)01368-8.

[2] A. Al-Mayah, J. Moseley, and K. K. Brock. Contact surface and material nonlinearity modeling of human lungs. *Phys. Med. Biol.*, 53:305–317, 2008. doi:10.1088/0031-9155/53/1/022.

[3] P. Andreo. Monte Carlo techniques in medical radiation physics. *Phys. Med. Biol.*, 36(7): 861–920, 1991. doi:10.1088/0031-9155/36/7/001.

[4] N. Aspert, D. Santa-Cruz, and T. Ebrahimi. Mesh: Measuring errors between surfaces using the hausdorff distance. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, volume I, pages 705–708, 2002. URL `http://mesh.epfl.ch`.

[5] M. J. Berger, J. H. Hubbell, S. M. Seltzer, J. Chang, J. S. Coursey, R. Sukumar, D. S. Zucker, and K. Olsen. XCOM: Photon cross sections database, 2010. URL `http://www.nist.gov/pml/data/xcom`.

[6] R. Birch, M. Marshall, and G. M. Ardan. Catalogue of spectral data for diagnostic X-rays. Scientific Report Series N°30, The Hospital Physicists' Association, London, UK, 1979. 143p.

[7] F. E. Boas and D. Fleischmann. CT artifacts: Causes and reduction techniques. *Imaging Med*, 4(2):229–240, 2012.

[8] W. Dilworth Cannon, Gregg T. Nicandri, Karl Reinig, Howard Mevis, and Jocelyn Wittstein. Evaluation of skill level between trainees and community orthopaedic surgeons using a virtual reality arthroscopic knee simulator. *The Journal of Bone & Joint Surgery*, 96(7):e57, 2014. ISSN 0021-9355. doi:10.2106/JBJS.M.00779.

[9] J. Dankelman, M. Wentink, C. A. Grimbergen, H. G. Stassen, and J. Reekers. Does virtual reality training make sense in interventional radiology? Training skill-, rule- and knowledge-based behavior. *Cardiovasc. Intervent. Radiol.*, 27(5):417–421, 2004. doi:10.1007/s00270-004-0250-y.

[10] S. L. Dawson, S. Cotin, D. Meglan, D. W. Shaffer, and M. A. Ferrell. Designing a computer-based simulator for interventional cardiology training. *Catheterization and Cardiovascular Interventions*, 51(4):522–527, 2000. ISSN 1522-726X. doi:10.1002/1522-726X(200012)51:4<522::AID-CCD30>3.0.CO;2-7.

[11] A.-L. Didier, P.-F. Villard, J.-Y. Bayle, M. Beuve, and B. Shariat. Breathing thorax simulation based on pleura physiology and rib kinematics. In *Proc. MediVis*, pages 35–42, 2007.

[12] M. Folkerts, X. Jia, X. Gu, D. Choi, A. Majumdar, and S. Jiang. Implementation and evaluation of various DRR algorithms on GPU. *Medical Physics*, 37(6):3367, 2010. doi:10.1118/1.3469159.

[13] D. Fortmeier, A. Mastmeyer, J. Schroder, and H. Handels. A virtual reality system for PTCD simulation using direct visuo-haptic rendering of partially segmented image data. *IEEE Journal of Biomedical and Health Informatics*, PP(99):1–1, 2014. ISSN 2168-2194. doi:10.1109/JBHI.2014.2381772.

[14] N. Freud, P. Duvauchelle, J. M. Letang, and D. Babot. Fast and robust ray casting algorithms for virtual X-ray imaging. *Nucl. Instrum. Methods Phys. Res., Sect. B*, 248(1):175–180, 2006. doi:10.1016/j.nimb.2006.03.009.

[15] S. F. F. Gibson. Linked volumetric objects for physics-based modeling. Technical Report TR97-20, Mitsubishi Electric Research Labs, 1997.

[16] J. H. Hubbell and S. M. Seltzer. Tables of x-ray mass attenuation coefficients and mass energy-absorption coefficients from 1 kev to 20 MeV for elements Z = 1 to 92 and 48 additional substances of dosimetric interest, 1996. URL `http://www.nist.gov/pml/data/xraycoef/`.

[17] F. Inanc, Joseph N. Gray, Terrence Jensen, and J. Xu. Human body radiography simulations: development of a virtual radiography environment. In *Proc. SPIE*, volume 3336, pages 830–837, 1998. doi:10.1117/12.317091.

[18] H. Ito, S. Koshizuka, R. Shino, A. Haga, T. Onoe, and K. Nakagawa. Quasi-4DCT images based on physics-based lung deformation simulation. *Radiother Oncol*, 99(Supplement 1): S484–S485, May 2011.

[19] L. K. R. Jahnke and et al. GPU acceleration of GEANT4 based monte carlo simulations for radio therapy. *Int. J. Radiation Oncology*, 72:S628, 2008.

[20] Koji Kawaguchi, Hiroyuki Egi, Minoru Hattori, Hiroyuki Sawada, Takahisa Suzuki, and Hideki Ohdan. Validation of a novel basic virtual reality simulator, the lap-x, for training basic laparoscopic skills. *Minimally Invasive Therapy & Allied Technologies*, 23(5):287–293, 2014. doi:10.3109/13645706.2014.903853.

[21] A. P. King, K. S. Rhode, Y. Ma, C. Yao, C. Jansen, R. Razavi, and G. P. Penney. Registering preprocedure volumetric images with intraprocedure 3-D ultrasound using an ultrasound imaging model. *IEEE Trans Med Imaging*, 29(3):924–937, March 2010. doi:10.1109/TMI.2010.2040189.

[22] E. Kyriakou, D. R. McKenzie, N. Suchowerska, and R. R. Fulton. Breathing as a low frequency wave propagation in nonlinear elastic permeable medium. In *Proc. ETOPIM*, pages 311–314, 2007.

[23] F. Levet, X. Granier, and C. Schlick. Fast sampling of implicit surfaces by particle systems. In *Proc. IEEE International Conference on Shape Modeling and Applications*, volume 00, page 39, 2006.

[24] Bruno Levy and Nicolas Bonneel. Variational anisotropic surface meshing with voronoi parallel linear enumeration. In Xiangmin Jiao and Jean-Christophe Weill, editors, *Proceedings of the 21st International Meshing Roundtable*, pages 349–366. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-33572-3.

[25] N. Li, S.-H. Kim, J.-H. Suh, S.-H. Cho, , J.-G. Choi, and M.-H. Kim. Virtual X-ray imaging techniques in an immersive casting simulation environment. *Nucl. Instrum. Methods Phys. Res., Sect. B*, 262(1):143–152, 2007. doi:10.1016/j.nimb.2007.04.262.

[26] Y. Li and K. Brodlie. Soft object modelling with generalised chainmail - extending the boundaries of web-based graphics. *Comput. Graph. Forum*, 22(4):717–727, 2003. doi:10.1111/j.1467-8659.2003.00719.x.

[27] Anthony E. Lujan, Edward W. Larsen, James M. Balter, and Randall K. Ten Haken. A method for incorporating organ motion due to breathing into 3D dose calculations. *Medical Physics*, 26(5):715–720, 1999. doi:10.1118/1.598577.

[28] R. M. H. McMinn. *Last's Anatomy: Regional and Applied.* Churchill Livingstone, 1990. ISBN: 0-443-03483-4.

[29] J.A. Milburn, G. Khera, S.T. Hornby, P.S.C. Malone, and J.E.F. Fitzgerald. Introduction, availability and role of simulation in surgical education and training: Review of current evidence and recommendations from the association of surgeons in training. *International Journal of Surgery*, 10(8):393–398, 2012. ISSN 1743-9191. doi:10.1016/j.ijsu.2012.05.005.

[30] M. P. Pato, N. J. Santos, P. Areias, E. B. Pires, M. de Carvalho, S. Pinto, and D. S. Lopes. Finite element studies of the mechanical behaviour of the diaphragm in normal and pathological cases. *Comput Methods Biomech Biomed Engin*, 14(6):505–513, 2011.

[31] A. P. Santhanam, C. M. Fidopiastis, P. Davenport, K. Langen, S. Meeks, and J. P. Rolland. Real-time simulation and visualization of subject-specific 3D lung dynamics. In *Proc. IEEE CBMS*, pages 629–634, 2006.

[32] W. Schneider, T. Bortfeld, and W. Schlegel. Correlation between CT numbers and tissue parameters needed for monte carlo simulations of clinical dose distributions. *Physics in Medicine and Biology*, 45(2):459–478, 2000.

[33] W. P. Segars, D. S. Lalush, and B. M. W. Tsui. Modeling respiratory mechanics in the MCAT and spline-based MCAT phantoms. *IEEE T. Nucl. Sci.*, 48(1):89–97, 2001.

[34] Varian Medical Systems, Inc. Radiographic rad/fluoro x-ray tubes, 2011. URL `http://www.varian.com/us/xray/products/immediate_delivery/radiographic_rad_fluoro.html`.

[35] Rickul Varshney, Saul Frenkiel, Lily HP Nguyen, Meredith Young, Rolando Del Maestro, Anthony Zeitouni, Marc A Tewfik, et al. Development of the McGill simulator for endoscopic sinus surgery: A new high-fidelity virtual reality simulator for endoscopic sinus surgery. *American journal of rhinology & allergy*, 28(4):330–334, 2014.

[36] F. P. Vidal, J. M. Létang, G. Peix, and P. Cloetens. Investigation of artefact sources in synchrotron microtomography via virtual X-ray imaging. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 234(3):333–348, June 2005. doi:10.1016/j.nimb.2005.02.003.

[37] F. P. Vidal, M. Garnier, N. Freud, J. M. Létang, and N. W. John. Simulation of X-ray attenuation on the GPU. In *Proc. of TPCG 2009*, pages 25–32, 2009.

[38] F. P. Vidal, M. Garnier, N. Freud, J. M. Létang, and N. W. John. Accelerated deterministic simulation of x-ray attenuation using graphics hardware. In *Eurographics 2010 - Poster*, page Poster 5011, 2010.

[39] P. Vidal, Franck, Pierre-Frédéric Villard, and Evelyne Lutton. Tuning of patient specific deformable models using an adaptive evolutionary optimization strategy. *IEEE Transactions on Biomedical Engineering*, 59(10):2942–2949, October 2012. doi:10.1109/TBME.2012.2213251.

[40] P. F. Villard, F. P. Vidal, C. Hunt, F. Bello, N. W. John, S. Johnson, and D. A. Gould. Percutaneous transhepatic cholangiography training simulator with real-time breathing motion. In *Proc. of CARS 2009*, pages S66–S67, 2009.

[41] P.-F. Villard, F. P. Vidal, C. Hunt, F. Bello, N. W. John, S. Johnson, and D. A. Gould. Simulation of percutaneous transhepatic cholangiography training simulator with real-time breathing motion. *International Journal of Computer Assisted Radiology and Surgery*, 4(9): 571–578, November 2009. doi:10.1007/s11548-009-0367-1.

[42] P. F. Villard, P. Boshier, F. Bello, and D. Gould. Virtual reality simulation of liver biopsy with a respiratory component. In *Liver Biopsy*, pages 315–334. InTech, 2011.

[43] M. von Siebenthal, Gàber Székely, A. Lomax, and Philippe Cattin. Inter-subject modelling of liver deformation during radiation therapy. In *Proc. MICCAI*, pages 659–666, 2007.

[44] T. A. Wilson, A. Legrand, P.A. Gevenois, and A. Troyer. Respiratory effects of the external and internal intercostal muscles in humans. *J. Physiol. (Lond.)*, 530(2):319–330, 2001.

[45] X. Wu, J. Allard, and S. Cotin. Real-time modeling of vascular flow for angiography simulation. In *Med. Image. Comput. Comput. Assist. Interv.*, volume 4791 of *Lecture Notes in Computer Science*, pages 557–565, 2007. doi:10.1007/978-3-540-75757-3_68.

[46] P. A. Yushkevich, J. Piven, C. Hazlett, G. Smith, S. Ho, J. C. Gee, and G. Gerig. User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. *Neuroimage*, 31(3):1116–1128, 2006.

[47] B. Zhu, L. Gu, and J. Zhang. A method for collision response between deformable objects in virtual surgery. In *Proc. ITAB*, pages 119–122, 2007.

[48] V. B. Zordan, B. Celly, B. Chiu, and P. C. DiLorenzo. Breathe easy: Model and control of simulated respiration for animation. In *Proc. ACM SIGGRAPH/Eurographics SCA*, pages 29–37, 2004.