Joint Simulation of Transmission X-ray Imaging on GPU and Patient's Respiration on CPU

Franck P. Vidal, Pierre-Frédéric Villard, Manuel Garnier, Nicolas Freud, Jean Michel Létang, Nigel W. John, Fernando Bello

▶ To cite this version:

Franck P. Vidal, Pierre-Frédéric Villard, Manuel Garnier, Nicolas Freud, Jean Michel Létang, et al.. Joint Simulation of Transmission X-ray Imaging on GPU and Patient's Respiration on CPU. American Association of Physicists in Medicine Annual Meeting - AAPM 2010, Jul 2010, Philadelphia, United States. hal-00485720

HAL Id: hal-00485720

https://hal.science/hal-00485720

Submitted on 20 Sep 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Joint Simulation of Transmission X-ray Imaging on GPU and Patient's Respiration on CPU

F. P. Vidal[1,2], P. F. Villard[3,4], M. Garnier[5,2], N. Freud[6],
J. M. Létang[6], N. W. John[2] and F. Bello[3]

[1]University of California, San Diego, CA

[2]Bangor University, UK

[3]Imperial College, London, UK

[4]LORIA - Nancy University, France

[5]BRGM, France

[6]INSA - Lyon, France

Abstract

In this extended abstract, we propose a join deterministic simulation of i) X-ray transmission imaging on the GPU and ii) patient's respiration on the CPU. It has been initially deployed into a medical simulator for training percutaneous transhepatic cholangiography (PTC), i.e. fluoroscopy guidance of needles. The patient's respiration is modelled and it supports different breathing patterns. The dynamic polygon meshes of the patient is then used as the input data for the X-ray simulation.

Our previous attempt to simulate X-ray attenuation from polygon meshes utilising the GPU showed significant increase of performance, with respect to a validated software implementation, without loss of accuracy. However, the simulations were restricted to monochromatic X-rays and finite point sources. We present here an extension to our method to perform physically more realistic simulations by taking into account polychromatic X-rays and focal spots causing blur.

1 Introduction

The simulation of transmission X-ray imaging is often performed deterministically (i.e. no interaction is taken into account) on the CPU using ray-tracing as a fast alternative to Monte Carlo methods [1].

In [3], we proposed to compute the X-ray attenuation from polygon meshes directly on the GPU using OpenGL and the OpenGL Shading Language (GLSL) to significantly increase performance without loss of accuracy. The method has been deployed into a medical simulator for training PTC [4]. It makes used of polygon meshes that are dynamically modified depending on the respiration cycle of the virtual patient. The simulations were however restricted to monochromatic X-ray beams and finite point sources.

We have now extended the simulation pipeline to take into account focal spots that cause geometric unsharpness and polychromatic X-rays (see Section 2) and dynamic polygon meshes can be used as input data (see Section 3).

2 Simulation Pipeline

The Beer-Lambert law relates the absorption of light to the properties of the material through which the light is travelling. For a polychromatic incident X-ray beam, it is:

$$N_{out}(E) = N_{in}(E) \exp\left(- \sum_{i=0}^{i<objs} \mu(E,i) L_p(i) \right) \quad (1)$$

with $N_{in}(E)$ the number of incident photons at energy E, $N_{out}(E)$ the number of transmitted photons at energy E. $objs$ is the total number of objects in the 3D scene. μ the linear attenuation coefficient (in cm^{-1}), which depends on: i) E, the energy of incident photons, and ii) the material properties of the object. $L_p(i)$ is the path length (in cm) of the ray in the i$^{\text{th}}$ object.

We originally adopted the algorithm presented by Freud *et al*[1] to store the length of a ray crossing a given 3D object (see [3] for details about the GPU implementation).

To implement the Beer-Lambert law with a polychromatic beam (Eq. 1), additional loops (see grey boxes in Fig. 1) have been included in our simulation pipeline. It is split into different rendering passes and uses frame buffer objects (FBOs) to store intermediate results. Fig. 1 shows the full pipeline. It takes into account both the geometrical blur (see Fig. 2) and polychromatic X-rays (see Fig. 3).

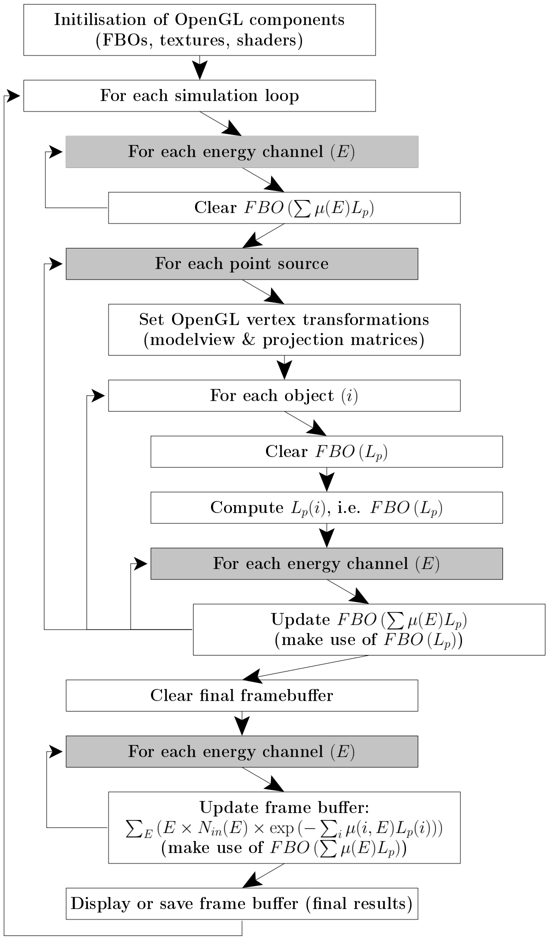
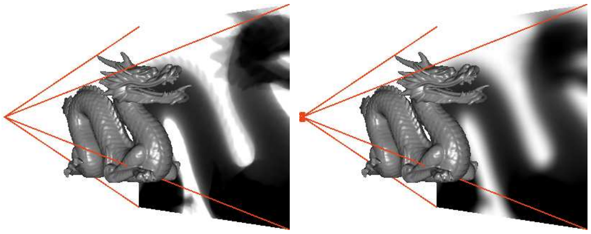

Initilisation of OpenGL components (FBOs, textures, shaders)
For each simulation loop
For each energy channel (E)
Clear $FBO\,(\sum \mu(E)L_p)$
For each point source
Set OpenGL vertex transformations (modelview & projection matrices)
For each object (i)
Clear $FBO\,(L_p)$
Compute $L_p(i)$, i.e. $FBO\,(L_p)$
For each energy channel (E)
Update $FBO\,(\sum \mu(E)L_p)$ (make use of $FBO\,(L_p)$)
Clear final framebuffer
For each energy channel (E)
Update frame buffer: $\sum_E (E \times N_{in}(E) \times \exp(-\sum_i \mu(i,E)L_p(i)))$ (make use of $FBO\,(\sum \mu(E)L_p)$)
Display or save frame buffer (final results)

Figure 1: Simulation pipeline.

The shape of the source is modelled using a variable number of point sources (see Fig. 2(b)). It can be a square, a cube, a sphere or any set of random 3D points. Each point is assigned a fraction of the total number of photons in the system.

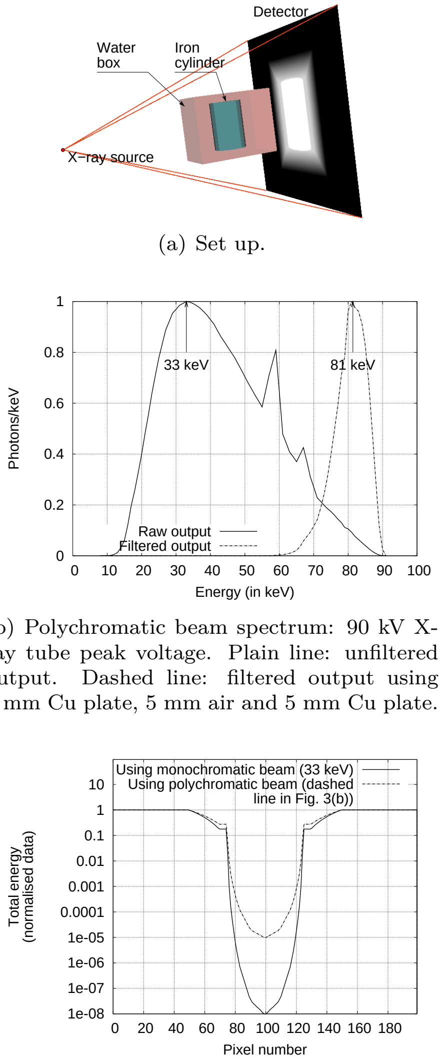
The incident beam is split into discrete energy channels. A 3D texture is now attached to the FBO used to store $\sum \mu(i, E)L_p(i)$. Each slice of the 3D texture corresponds to an energy channel (E). To produce the final image, the total amount of energy received by each pixel is computed. Fig. 3(c) shows profiles produced from a simple geometrical set-up (see Fig. 3(a)) and using a monochromatic beam (33 keV) and a polychromatic beam (see plain



(a) Without geometric unsharpness.

(b) With geometric unsharpness using a square source (1 cm^2).

Figure 2: Simulations with different source sizes.



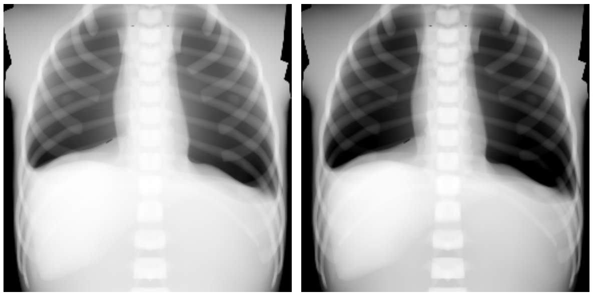
(a) Set up.

(b) Polychromatic beam spectrum: 90 kV X-ray tube peak voltage. Plain line: unfiltered output. Dashed line: filtered output using 5 mm Cu plate, 5 mm air and 5 mm Cu plate.

(c) Centre line profiles.

Figure 3: Incidence of the beam spectrum.

line in Fig. 3(b)). Fig. 4 shows images simulated using different polychromatic beams and a realistic patient model. Resulting images are first normalised between 0 and 1, then displayed using the same visualisation window.



(a) Using the unfiltered beam. (b) Using the filtered beam.

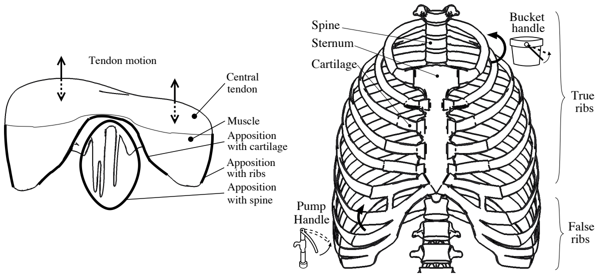
Figure 4: X-ray simulations using polychromatic X-rays (see beams in Fig. 3(b)).

3 Respiration Modelling

We saw in the previous section that polygon meshes are used as input data for our X-ray simulation pipeline. Therefore, dynamic data mimicking the patient's respiration can be used. Our respiration model is based on a deep study of the thorax anatomy and physiology. It consists in a combination of rib motion following a kinematic law, a translation of the diaphragm tendon. The soft tissue deformation is simulated using the ChainMail algorithm (see [4] for further details).

The Respiratory System: Respiration is a physiological process involving different muscles. It can be separated into two main contributions: i) the thorax motion with the intercostal muscles allowing the ribs to expand, and ii) the diaphragm action whereby the diaphragm muscles contract or relax during the breathing cycle.

The diaphragm (Fig. 5(a)) can be considered as a heterogeneous sheet of tendon and muscle. The circumferential part is muscular and is fixed to the lowest ribs. During the inspiration, the muscular fibres contract and draw the central tendon downwards and forwards. In this movement, the curvature of the diaphragm is barely altered, the dome moving downwards nearly parallel to its original position. The central tendon, applied to the abdominal viscera, then becomes a fixed point for the action of the diaphragm, the effect of which is to elevate the lower ribs and through them to push forwards the body of the sternum and the upper ribs. The thoracic skeleton is mainly composed of ribs (Fig. 5(b)). The first seven ribs, also called true ribs, are connected to the spinal column and, via costal cartilages, to the sternum. The five remaining ribs - or false ribs - are composed by three ribs whose cartilages are attached to those of the rib above. The space between two ribs is called intercostal space and contains intercostal muscles causing their motion during respiration. The movement of the ribs can be considered as the combination of a "bucket handle" movement and a "pump handle" movement. The "bucket handle" movement increases the lateral excursion of the ribs, while the "pump handle" one increases the anteroposterior diameter of the thorax. Thus, the rib cage diameter increases in both the anteroposterior and the transverse directions.



(a) Diaphragm heterogeneous composition. (b) Rib cage with the two kinds of rib motion.

Figure 5: Thorax anatomy.

Respiration Model Algorithm: As explained in Section 3, the rotation of the ribs can be defined by the combination of two different rotations: the "bucket handle" and the "pump handle" movement. They are respectively defined here by the angles α and β. Wilson *et al* [2] measured these angles for five subjects at Functional Residual Capacity (FRC) and Total Lung Capacity (TLC). The rib rotation was thus decomposed into two distinct rotations. We consider here that the rest position is at FRC. In our model, the rotation angles are given by the following equations:

$$\alpha' = ((\alpha_{\mathrm{TLC}} - \alpha_{\mathrm{FRC}}) \times \tfrac{1}{2} \times (1 + sin(2\pi f t)) \quad (2)$$

$$\beta' = ((\beta_{\mathrm{TLC}} - \beta_{\mathrm{FRC}}) \times \tfrac{1}{2} \times (1 + sin(2\pi f t)) \quad (3)$$

where: α' is the reduced "pump handle" angle $\alpha'(\mathrm{FRC}) = 0$ and $\alpha'(\mathrm{TLC}) = \alpha_{\mathrm{TLC}} - \alpha_{\mathrm{FRC}}$, β' is the reduced "bucket handle" angle $\beta'(\mathrm{FRC}) = 0$ and $\beta'(\mathrm{TLC}) = \beta_{\mathrm{TLC}} - \beta_{\mathrm{FRC}}$, f :is the respiration frequency and t is the time.

Although the ribs were modelled as a rigid body, such assumption is not possible for the diaphragm. The method used for the deformation modelling is based on the 3D ChainMail algorithm extension proposed by Li and Brodlie [5]. In this model, mesh elements are interconnected as links in a chain. So, within a certain limit, each point can move freely without influencing its neighbours. When one 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. The ChainMail parameters to simulate the deformation behaviour include the compression, the stretching and the shearing. To model the central tendon as a tendon and not as a muscle, it has to be almost rigid. This can be done by setting the ChainMail parameters to specific values. When the compression parameter is equal to 1, the stretch parameter is equal to 1 and the shear parameter is equal to 0, the object is rigid. To do a vertical movement, one central point is forced to have a sinusoidal movement along the vertical axis similarly to Equations 2 and 3. During inhalation, the central tendon has a downward movement, which is synchronous with the expanding rib cage. As the links for the central tendon are rigid, the points corresponding to this tendon will follow the exact same movement. The other points will have the expected "chain reaction" according to the ChainMail rules. At FRC, the diaphragm has a shape like what can be seen in Fig. 5(a). Thus, it must be able to compress itself during inhalation. The chosen value for the compression parameter is 0.7. The stretch and shear components are not very important here. The value for the stretch parameter and the shear parameter are 1.1 and 0.1, respectively. However, the diaphragm is also supposed to follow the movement of the ribs. To do, this the idea is to "fix" some points of the diaphragm to some points of the ribs. Finally, the movement of the diaphragm will be a combination of the downward movement of the central tendon and the movement induced by the contact points with the ribs. An illustration of both thoracic and diaphragmatic motion could be seen in Fig. 6.

4 Conclusion

X-ray transmission images can be fully simulated on the GPU, by using the Beer-Lambert law with polychromatism and taking into account the shape of the source. The respiration of the patient can be modelled to produce dynamic meshes. This is a useful development to improve the level of realism in simulations, when both speed and accuracy have to be retained, e.g. in radiotherapy online replanning.

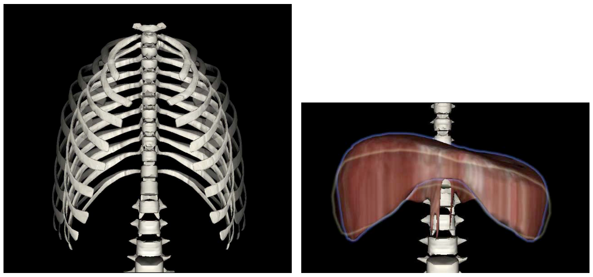

(a) Rib cage. (b) Diaphragm.

Figure 6: Respiration simulation: expiration to inspiration (transparent rendering).

References

[1] N. Freud, P. Duvauchelle, J. M. Létang, and D. Babot. Fast and robust ray casting algorithms for virtual X-ray imaging. *Nucl Instrum Methods Phys Res B*, 248(1):175–180, 2006.

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

[3] 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 Theory and Practice of Computer Graphics 2009*, pages 25–32, 2009.

[4] 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. *Int J Comput Assist Radiol Surg*, 4(9):571–578, 2009.

[5] L. Ying and K. Brodlie. Soft object modelling with generalised ChainMail - extending the boundaries of web-based graphics. *Comput Graph Forum*, 22(4):717–727, 2003.