Modeling Air-flow in the Tracheobronchial tree using Computational Fluid Dynamics

Ilhan Kaya[1], Anand P. Santhanam[2], Celina Imielinska[3] and Jannick Rolland[1,2]

[1] School of Computer Science and Engineering
[2] CREOL, College of Optics and Photonics
University of Central Florida
4000 Central Florida Blvd Orlando, FL 32816
ikaya@creol.ucf.edu
[3] Department of Biomedical Informatics, Columbia University,
701 West 168th Street
New York, NY 10032

Abstract. In this paper, we present a biomechanical framework to model air-flow inside the bronchus and deformations across the tracheobronchial tree, pipeline for the simulator, theory and initial steps to realize this framework on a highly parallel graphical processing unit (GPU). We discuss the main challenges expected and encountered to date. By using computational fluid dynamics (CFD) and computational solid dynamics (CSD) principles, we propose a numerical simulation framework that includes a biomechanical model of the tracheobronchial tree to simulate air flow inside the tree, on GPU in real-time. The proposed 3D biomechanical model to simulate the air inside the lungs coupled with a deformation model of the tracheobronchial tree, expressed through fluid-structure interaction, can be used to predict the transformations of the voxels from a 4D computed tomography (4DCT) dataset. Additionally, the proposed multi-functional CFD and CSD based framework is suitable for clinical applications such as adaptive lung radiotherapy, and a regional alveolar ventilation estimation.

1 Introduction

Lung cancer and various other ongoing or chronic lung diseases, e.g. closure of airways and emphysema, if lumped together, it is the number three killer in the United States. There has been a significant amount of research in analyzing the tissues of the lung through advanced imaging modalities such as Positron Emission Tomography (PET), Computerized Tomography (CT), and Magnetic Resonance Imaging (MRI). By applying various forms of medical image processing algorithms to CT, MRI, and PET images, anatomical structures and their material properties can be explored. Tawhai et al. make use of various forms of image segmentation algorithms to generate subject-specific computational meshes of the human bronchial tree [1, 2]. Hoffman et al. used deformable registration algorithms on CT scans to estimate the regional ventilation in ovine lungs [3, 4]. In the past few years, researchers all over the

world have been able to simulate natural phenomena such as fluids, gases, fire behavior on GPUs accurately and faster than CPUs. Three different approaches to simulate fluids in the field of computational fluid dynamics exist, fixed grid-based (Euler) or particle motion (Lagrange) and a hybrid Arbitrary Lagrangian-Eulerian (ALE). Eulerian methods evaluate material properties at stationary grid points, whereas Lagrangian methods employ moving grids with the material particles to solve the governing equations [5]. Kruger et al. built up a million Lagrangian particles to simulate and visualize 3D flow fields on non-uniform grids [6]. Dynamically regenerated meshes are used in [7] to simulate fluids. Viscoelastic material simulation based upon Eulerian methods is shown in [8]. Harris et al. make use of GPU to simulate cloud dynamics whose state is governed by incompressible Navier Stokes (NS) equations [9, 10].

In this paper, we propose a simulation framework to simulate the air inside the tracheobronchial tree on GPU. As part of this framework, we propose a 3D biomechanical model which accounts for the air motion inside the tracheobronchial tree, and the deformations on the tracheobronchial tree. This work is the first of its kind to simulate elasticity theory and fluid dynamics to predict the displacements and deformations of the human tracheobronchial tree on GPU, which provides opportunity to run the simulation in real-time and on customer PCs. The framework and biomechanical model proposed might be used to account for the alveoli ventilation when it is combined with an image registration algorithm, and it can also account for transport of aerosol contained in cigarette smoking in human lung.

2 GPU Pipeline

GPUs are very powerful, highly parallel stream processing processors. Since they are specifically built for vector processing, they have a parallel computer architecture model. Their architecture is built upon single instruction multiple data (SIMD) paradigm. In SIMD, a single node dispatches the instruction to other processing nodes that compute the instruction with their own local data. SIMD computers provide superiority over instruction driven processors, CPUs, due to their parallel execution of the instruction.

In Fig. 1 there is an illustration of floating point computational power of GPUs with respect to CPUs. This superiority makes GPUs an ideal platform to solve data-driven numerical problems in parallel. In a typical GPU, which is available on almost any PC sold today in market; there are one vertex and one fragment processor. Each processor is realized through several pipelines, working in parallel. In a NVIDIA GTX8800, which we use for our framework, there are 128 vertex and fragment processors pipelines. In Fig. 2, the GPU pipeline that realizes the stream processing nature of these parallel machines is illustrated.

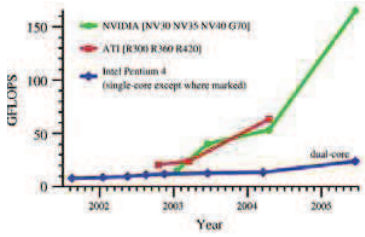

Fig. 1. Illustration of 32-bit floating point GPU comparison courtesy of Engel et al [11].

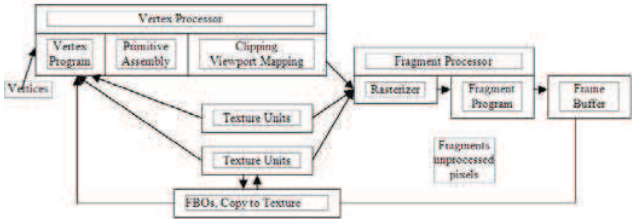

Fig. 2. Graphical Processing Pipeline, GPU Architecture

3 Biomechanical Modeling of Air inside the Tracheobronchial Tree and its Deformations

The tracheobronchial tree geometry, shape, measurements and dimensions change for each individual. Average tracheal length ranges from 5.4 cm to 13.1 cm, as the age of the subjects changes from 2 to 20 [12]. Generally, air flow through the terminal bronchioles follows a laminar pattern, whereas air flow inside the trachea can be turbulent with high ventilation rates in abnormal conditions [13]. Hence our biomechanical model consists of two parts to be able to correctly model the physics of the tracheobronchial tree: Modeling air inside the tracheobronchial tree, and modeling the deformations on the tracheobronchial tree.

3.1 Modeling air trough computational fluid dynamics (airway gas dynamics)

We model air inside the tracheobronchial tree as a fluid defined as any material which deforms continuously under shear stress. We make use of the continuum assumption of fluid mechanics, which states that fluid can be treated as an infinitely divisible substance. We divide the solution domain into 3D infinitesimal cells, where the velocity and the density of the fluid are defined as average properties. Our modeling assumptions for fluid is being incompressible, i.e. each cell has constant volume over time, homogeneous, density of the fluid stays constant in space, viscous, rate of deforma-

tion is same under the same stress for all fluid cells. Fluid cells might experience deformation, rotation, translation in space. The state of the fluid is determined through velocity of the each fluid cell, being a function of time and space, effectively represented as a vector field, and pressure of each fluid cell, which is function of space and time.

3.1.1 Equations of Fluid State

Governing equations for fluid motion are the Navier-Stokes equations (NS). The state of the fluid, given by velocity and pressure, can be determined through solutions of NS.

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \bullet \nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + \upsilon \nabla^2 \mathbf{u} + \mathbf{F} \quad (1)$$

$$\nabla \bullet \mathbf{u} = 0 \quad (2)$$

In the above equations, ***u*** represents velocity, a vector quantity, having 3 components in 3D. If ***x****(x,y,z)* represents the position of the fluid cell, the velocity is ***u****(**x**,t)* and the pressure is *p(**x**,t)*. ρ represents the density of the fluid, υ represents the kinematics viscosity. ***F****(**x**,t)* represent the external forces acting on the fluid including the gravitational force. The $(\mathbf{u} \bullet \nabla)\mathbf{u}$ term is called the advective term and it is the source of nonlinearity. It resembles the propagation of any disturbance caused by the external forces acting on the fluid. The $\frac{1}{\rho}\nabla p$ term is the effect of pressure on the rate of change of the velocity. The $\upsilon \nabla^2 \mathbf{u}$ term represents viscous diffusion. Due to viscosity, fluids are resistive to move, this resistance results in the distribution of momentum. Eq. 2 states the mass conversation since the fluid is assumed to be incompressible. As for the boundary conditions to solve NS, we use no-slip velocity and pure Neumann pressure condition. No-slip velocity states that velocity is zero in all 3 dimensions at the boundaries. Pure-Neumann boundary conditions state that the rate of change of pressure is zero along the normal direction on the surface boundaries.

3.1.2 A Solution Method for Equations of Fluid State

The solution of NS, the set of Partial Differential Equations (PDE), can be found in many texts. We follow the lead of Griebel et al. [14] and Marsden et al. [15] Marsden et al. show the solution of the NS through a theorem in vector calculus. This theorem is called the Helmholtz-Hodge Decomposition Theorem. It states that in the same way a vector is decomposed into two components, a vector field can be decomposed into two vector fields. A divergent vector field **w** can be decomposed into a non-divergent vector field **u** and a vector field composed by the gradient of the scalar field. The Helmholtz Hodge decomposition theorem is shown in Eq. 3.

$$\mathbf{w} = \mathbf{u} + \nabla p \quad (3)$$

Since the application of external forces, advection and viscous diffusion, results in a divergent velocity field, $\boldsymbol{w}$, at the end of each time step, we must satisfy the continuity requirement by subtracting the gradient of pressure field from the divergent velocity field.

$$\mathbf{u} = \mathbf{w} - \nabla p \tag{4}$$

If we apply divergence to both sides of Eq 3, then we have

$$\nabla \bullet \mathbf{w} = \nabla \bullet \mathbf{u} + \nabla \bullet \nabla p$$

$$\nabla \bullet \mathbf{w} = \nabla^2 p \tag{5}$$

Since $\mathbf{u}$ is a divergent-free vector field, $\nabla \bullet \mathbf{u}$ drops out. Eq. 5 is known as the Poisson-pressure equation. Once we have the pressure distribution, Eq. 4 gives the velocity at each time t, for any point in the fluid domain. We can further simplify Eq. 1 by defining a projection operator, $\boldsymbol{\Omega}$, which projects the divergent vector field $\boldsymbol{w}$ into its non-divergent vector field, $\mathbf{u}$. If we apply $\boldsymbol{\Omega}$ to both sides of Eq. 3,

$$\Omega(\mathbf{w}) = \Omega(\mathbf{u}) + \Omega(\nabla p) \tag{6}$$

By definition of $\boldsymbol{\Omega}$

$$\Omega(\mathbf{w}) = \Omega(\mathbf{u}) = \mathbf{u} \tag{7}$$

If we substitute Eq. 7 into Eq. 6

$$0 = \Omega(\nabla p) \tag{8}$$

Applying the projection operator $\boldsymbol{\Omega}$ to Eq. 1 and substituting Eq. 8 into the resulting formulation will yield a drop out of the pressure term from Eq. 1.

$$\frac{\partial \mathbf{u}}{\partial t} = \Omega\left[-(\mathbf{u} \bullet \nabla)\mathbf{u} + \upsilon \nabla^2 \mathbf{u} + \mathbf{F}\right] \tag{9}$$

3.2 Modeling tracheobronchial wall dynamics using linear elasticity theory

An important characteristic of the tracheobronchial tree is that it deforms constantly under shear stress due to its elastic nature. When the fluid starts to move inside the tracheobronchial tree, its pressure distribution causes displacements among other forces acting on the tracheobronchial tree. During an inhalation expiration cycle, changes in the transmural pressure inside and outside of the tracheobronchial tree causes deformations on the walls. These deformations furthermore change the boundaries for the fluid motion, resulting in displaced boundaries for the fluid simulation. This type of coupling between fluid and an elastic wall is modeled through Fluid-Structure Interaction (FSI). Fluid pressure distribution gives the force exerted upon the elastic wall, whereas elastic displacements calculated through elastic body theory of solid mechanics, give the new boundaries for fluid. Holzhauser et al. developed a mathematical model to estimate the change in cross-sectional area of the trachea based upon a coupling between fluid flow and elastic theory [16]. FSI models

are being also used in the simulation of blood flow in highly deformable arteries. In [17], authors developed a method to couple elastic arteries wall deformation as a moving boundary condition for blood flow. Equations for the incompressible fluid flow will be the same as given in Eq. 1. and Eq. 2. However the boundaries for the simulation for each time step will be redefined with the displacement calculated through the elasticity theory. Based on a staggered partition approach, we will divide the problem domain into two steps as shown by Griebel et al. [18]. At each time step, the fluid flow will be solved as if it has fixed boundaries, and the pressure distribution calculated by the fluid flow will be an input to the elasticity problem. Then newly calculated displacements through elasticity will be fed into fluid flow as new boundaries. Fig 3 gives an overview of FSI.

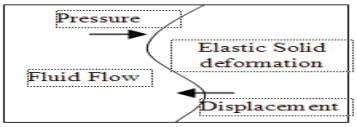

Fig. 3. A schematic of Fluid-Structure Interaction Modeling

In the elastic body deformation domain, time dependent Lame Equation serve as the mathematical formulation of the elasticity problem. It is given as

$$\rho_s \frac{\partial^2 u}{\partial t^2} - \mu \Delta u - (\lambda + \mu) \nabla \nabla \bullet u = \rho_s b \tag{10}$$

where u represents the displacement of elastic solid, ρ_S is the density of elastic solid, b are the external forces acting on the elastic solid, μ and λ are material specific Lame constants. It is to be noted that the b includes the force caused by the fluid flow. Finite methods have been extensively investigated for computing the solution of the elastic equation.

3.3 Biomechanical Model

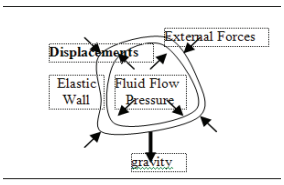

Fig. 4. 2D cross-section of a 3D biomechanical model of the human tracheobronchial tree.

In Fig. 4, a 2D cross-section of our biomechanical model for the tracheobronchial tree is shown. This model utilizes a fluid flow model of CFD with an elasticity theory of CSD in a FSI. Inputs to this model is gravitational force, external forces acting on the tracheobronchial tree, volume-time curve of the subject inhalation exhalation cycle, computational geometry of tracheobronchial tree, inhaled air parameters, i.e viscosity,

density, tissue properties for elasticity, density of the tissue, lame constants. As an output of this model, tracheobronchial deformations can be predicted as displacements.

4 Framework Pipeline

To implement the proposed framework, a simulation pipeline was developed, which is illustrated in Fig. 5. Acquisition of the geometry for the tracheobronchial tree is done through segmentation of the images obtained through CT. Fluid simulation is mostly done on GPU with an Eulerian grid approach where advection is done through a semi-Lagrangian method based on [19]. Elastic body simulation will be done through the Arbitrary Lagrangian-Eulerian (ALE) method, in process of implementation. Once we have the deformations of the tracheobronchial tree, 4DCT registration algorithm can run to estimate alveoli ventilation.

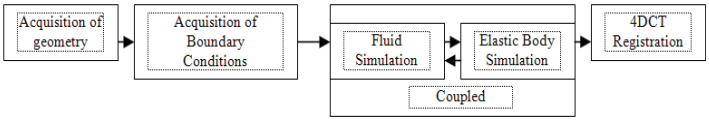

Fig. 5. Framework Pipeline to realize our 3D biomechanical model

5 Initial Implementation & Results

We have implemented initial operators which form the basis functions for the biomechanical framework proposed. The operators, addition, multiplication, gradient, divergence, and Jacobian are implemented on GPU. We use OPENGL and GLSL shading language to implement our framework. 32-bit floating point 3D textures are used to represent 3D uniform grid that we need to use for our discretization of the partial differential equation domain. As a discretization method, we use finite differences. 32-bit floating point textures are necessary to do any floating point arithmetic on GPU. In each channel of the 3D RGBA or luminance texture, the texture contains a true single precision floating point number that makes them suitable to represent 3D vector fields, i.e. velocity of fluid $\boldsymbol{w(x,t)}$, $\boldsymbol{u(x,t)}$, or scalar fluid pressure field, $p(\boldsymbol{x},t)$. To implement the general purpose calculation on GPU with 32-bit 3D textures, we render into 3D texture with frame buffer objects to have a feedback loop between framebuffer and texture units illustrated in Fig. 2. To compute each operation, we need to be able to generate as many fragments as the grid dimension. We render a dummy quadrilateral of filling the viewport for each slice of the 3D texture and run the computation on each fragment. Results are stored back on the 3D texture voxels trough framebuffer objects. Multi-texturing is used when there is more than one operand as in the case of addition. The gradient operator takes a 3D luminance texture as input and gives a 3D RGBA texture as output, since gradient operates on a scalar field and gives the greatest rate of change on that scalar field as a vector field. The gradient of the pressure field, $p(\mathbf{x},t)$, is calculated trough the finite difference formula given in

Eq. 11. We calculate the gradient of the pressure field to guarantee that the fluid obeys the divergent-free velocity assumption (incompressibility). Once the fluid is perturbed by an external force, the result is a divergent velocity field. This divergence on velocity field can be removed by subtracting the gradient of the pressure field at that time instant from the divergent velocity field according to the HelmHoltz Hodge theorem (Eq. 4).

$$\nabla p(\mathbf{x},t)=\left[\frac{\partial p}{\partial x},\frac{\partial p}{\partial y},\frac{\partial p}{\partial z}\right] \quad \frac{p_{i+1,j,k}-p_{i-1,j,k}}{2\delta x},\quad \frac{p_{i,j+1,k}-p_{i,j-1,k}}{2\delta y},\frac{p_{i,j,k+1}-p_{i,j,k-1}}{2\delta z} \tag{11}$$

The divergence operator takes a 3D velocity field, and gives a scalar field, so the input is a 3D RGBA texture, and the output is a luminance texture. Divergence of the velocity field is calculated trough the following finite difference formula.

$$\nabla \bullet \mathbf{u}=\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z} \quad \frac{u_{i+1,j,k}-u_{i-1,j,k}}{2\delta x}+\frac{v_{i,j+1,k}-v_{i,j-1,k}}{2\delta y}+\frac{w_{i,j,k+1}-w_{i,j,k-1}}{2\delta z} \tag{12}$$

For both the gradient and the divergence operators, we sample neighboring top, bottom, front, back, left, right voxels at a position in the grid in a GLSL shader and calculate the formulas presented with uniform grid dimension assumptions. Two Poisson equations arise in the solution of fluid equations, the viscous diffusion and the Poisson pressure equations. Once the fluid has been perturbed by an external force, the result is a divergent velocity field. We can obtain the pressure distribution by solving the Poisson pressure equation given in Eq. 5. In order to account for the viscosity of the fluid, we need to solve viscous diffusion equation, given in Eq. 13.

$$\frac{\partial \mathbf{w}}{\partial t}=\upsilon\nabla^2\mathbf{w} \tag{13}$$

Poisson equations can be solved by many numerical iteration methods i.e conjugate gradient, Gauss-Seidel, multi-grid, and Jacobian which are generally referred to as relaxation methods. We use the Jacobian due to its relatively easy implementation. By using the finite difference method, we expand the Laplacian operator, as it is given in Eq. 14.

$$\nabla^2 p(\mathbf{x},t)=\left[\frac{\partial^2 p}{\partial x^2}+\frac{\partial^2 p}{\partial y^2}+\frac{\partial^2 p}{\partial z^2}\right] \quad \frac{p_{i+1,j,k}-2p_{i,j,k}+p_{i-1,j,k}}{(\delta x)^2}+\frac{p_{i,j+1,k}-2p_{i,j,k}+p_{i,j-1,k}}{(\delta y)^2}+\frac{p_{i,j,k+1}-2p_{i,j,k}+p_{i,j,k-1}}{(\delta z)^2} \tag{14}$$

Once we substitute Eq. 14 into Eq. 5, then we obtain the iteration equation for the Jacobi algorithm, which gives us the pressure distribution across the fluid.

$$p_{i,j,k}^{n+1}=\frac{p_{i+1,j,k}^{n}+p_{i-1,j,k}^{n}+p_{i,j+1,k}^{n}+p_{i,j-1,k}^{n}+p_{i,j,k+1}^{n}+p_{i,j,k-1}^{n}-(\partial x)^2\nabla\bullet\mathbf{w}}{6} \tag{15}$$

We implemented these operators both on GPU and CPU, and plotted the frame per second (FPS) values versus the grid size as a comparison metric in Fig. 6. As it can be seen from Fig. 6, initially, when the grid dimension is small, CPU outperforms GPU, since it is designed to do general purpose computation, and setting up the fragment

pipeline on GPU takes up time. Once the grid dimension becomes 32 or higher, GPU outperforms CPU 10-30 times, being similar to the plot given in Fig. 1.

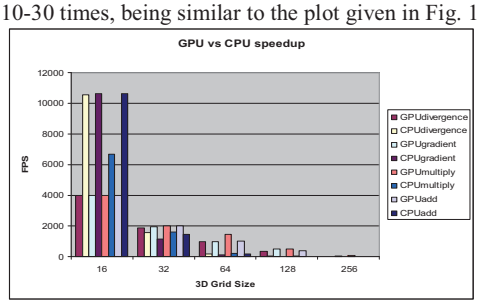

Fig. 6. GPU vs. CPU FPS comparison of the framework operators

6 Challenges

Poulikakos et al. have shown general challenges encountered while applying CFD methods to simulate fluids inside the human body, i.e. blood, cerebrospinal fluid [20]. In a similar way, we now list the challenges in implementing our biomechanical model. The complex geometry of the tracheobronchial tree makes the image acquisition and processing steps tedious. Due to the coupling of the elasticity problem with the fluid flow, fluid boundaries are modified at each time step, so traction of the boundaries is required. Hence boundary conditions are the Achilles's hill of the problem. As for the elasticity problem, material properties i.e. lame constants, density of tracheobronchial wall tissue, need to be specified. For the external forces acting on the exterior of the elastic tracheobronchial tree, an inverse computational dynamics approach needs to be specified. Numerical simulation of the problem requires discretization. For the fluid flow we use a finite-difference scheme. This is simple yet not suitable for fluid-structure problem. In order to solve Poisson Equations, Jacobi's iteration is used. Convergence and stability of numerical methods used in implementation should be verified.

7 Conclusion

In this paper we present a biomechanical framework to simulate air motion inside the tracheo-bronchial tree and estimate the deformations of the tracheobronchial tree caused by this fluid flow. This framework is being realized on GPU. The biomechanical model combines a fluid flow model based upon CFD with an elastic model of CSD. Furthermore, a simulator pipeline is given in this paper. Initial implementation results have been shown and challenges discussed.

Acknowledgements. This work is supported by US Army Medical Research and Material Command.

References

[1] Tawhai M. H., Hunter P.,Tschirren J., Reinhardt J.,McLennan G., Hoffman E.A.:CT-based Geometry Analysis and Finite Element Models of the Human and Ovine Broncial Tree. J. Appl. Physiol. Vol. 97 (2004) 2310-2321.

[2] Tawhai M. H., Pullan A.J., Hunter P.J.: Generation of an Anatomically-Based Three Dimensional Model of the Conducting Airways. Annals of Biomechanical Engineering. Vol 28 (2000) 793-802.

[3] Pan Y., Kumar D., Hoffman E.A., Christensen G.E., McLennan G., Song H.J., Ross A., Simon B.A., Reinhardt J.: Estimation of Regional Lung Expansion via 3D image Registration. Proceedings of SPIE Vol. 5746. Bellingham, USA (2005).

[4] Tawhai M. H., Nash M.P., Hoffman E.A.:A Imaging Based Computational Approach to Model Ventilation Distribution and Soft-tissue Deformation in the Ovine Lung. Academic Radiology. Vol. 13 (2006) 113-120.

[5] Feldman B.E., O'Brian J.F., Klingner B.M.,Goktekin G.E.: Fluids in Deforming Meshes. Proceedings of 2005 ACM SIGGRAPH/Eurographics Symposium. on Computer Animation. Los Angeles, USA (2005).

[6] Kruger J., Kipfer P., Kondratieva P.,Westermann R.: A particle System for Interactive Visualization of 3D Flows. IEEE T. on Visualization and Computer Graphics. Vol. 11 (2005) 744-756.

[7] Klingner B.M., Feldman B.E., Nuttapong C., O'Brian J.F.: Fluid Animation with Dynamic Meshes. Proceedings of 2006 ACM SIGGRAPH. Boston, USA (2006).

[8] Goktekin G. T., Barteil W. A., O'Brain J.F.: A Method for Animating Viscoelastic Fluids. Proceedings of 2004 ACM SIGGRAPH. Los Angeles, USA (2004).

[9] Harris J. M., Baxter V. W., Scheuermann T.,Lastra A.: Simulation of Cloud Dynamics on Graphics Hardware. Proceedings of 2003 ACM SIGGRAPH / Eurographics Conference on Graphics Hardware. San Diego, USA (2003).

[10] Harris J.M.: Fast Fluid Dynamics Simulation on the GPU. In GPU Gems Programming Techniques, Tips, Tricks for Real-time Graphics, Fernondo R. (Ed) Addison-Wesley Professional Boston (2004) 637-665.

[11] Engel K., Hadwiger M., Kniss M.J., Rezk-Salama C., Weiskopf D.:Real-Time Volume Graphics. A. K. Peters Ltd, Wellesley, USA (2006).

[12] Griscom N.T., and Wohl M.E.: Dimensions of the Growing Trachea Related to Age and Gender. American Journal of Roentgenology. Vol 146 (1986) 233-237.

[13] West J.B.: Respiratory Physiology- The Essentials (7_{th} Ed). Lippincott Williams & Wilkins (2005).

[14] Griebal M., Dornseifer T., NeunHoeffer T.: Numerical Simulation in Fluid Dynamics- A practical Introduction. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, USA (1998).

[15] Chorin A.J., and Marsden J.E.: A Mathematical Introduction to Fluid Mechanics (4_{th} Ed). Springer, (2000)

[16] Holzhauser U. and Lambert R.K.: Analysis of Tracheal Mechanics and Applications. J. Appl. Physiol. Vol. 91 (2006) 290-297.

[17] Figueroa A., Vignon-Clementel I., Jansen K.E., Hughes J.R.T., Taylor C.A.: A Coupled Momentum Method for Modeling Blood Flow in Three-Dimensional Deformable Arteries. Computer Methods in Applied Mechanics and Engineering. Vol. 195 (2006) 5685-5706

[18] Engel M., and Griebel M.: Flow Simulation on Moving Boundary-fitted Grids and Application to Fluid-Structure Interaction Problems. Int. J. Numerical Meth. In Fluids. Vol. 50 (2006) 437-468

[19] Stam J.:Stable Fluids. Proceedings of 1999 ACM SIGGRAPH. Los Angeles, USA (1999).

[20] Paulikakos D., Kurtcuoglu V.: Principles and Challenges of Computational Fluid Dynamics in Medicine. In 9^{th} International MICCAI Conference, Copenhagen, Denmark (2006).