3D Lung Dynamics in a Programmable Graphics Hardware for Augmented Reality based Medical Visualization

Anand P Santhanam[1], Felix G Hamza-Lup[1], Cali M Fidopiastis[2], Jannick P Rolland[123]

[1]*University of Central Florida, School of Computer Science,*
[2]*University of Central Florida, Institute for Simulation and Training,*
[3]*University of Central Florida, College of Optics and Photonics/FPCE,*
4000 Central Florida Blvd,Orlando 32816, USA

Abstract

Medical simulations of physically-based lung morphology constitute promises as effective tools for teaching and training clinical and surgical procedures related to lungs. Their effectiveness may be greatly enhanced when visualized in an Augmented Reality (AR) environment. However the computational requirements of AR environments limit the availability of Central Processing Unit (CPU) for the lung dynamics simulation under different conditions. In this paper we present a method for real-time lungs deformation simulation that takes advantage of the programmable Graphics Processing Unit (GPU), saving the CPU time for other AR associated tasks (e.g. tracking, communication and interaction management). We extend a method previously developed for simulations of the 3D lung dynamics using Green's formulation in the case of the upright position, to other patient's orientation as well as the subsequent changes in the lung dynamics. Specifically, the proposed method presents a computational optimization and its implementation in a GPU. Results show that the computational requirements for simulating the deformation of a 3D lung model are significantly reduced.

Key words: Lung Physiology, Spherical Harmonics, Green's function

1. Introduction

Medical simulation of respiratory physiology allows the development of promising tools for applications ranging from teaching and training to surgical guidance. Of particular importance is the simulation and visualization of physically-based lung dynamics based on the lung morphology. A medical evaluation of regional lung structure and its associated behavior during breathing, aids in estimating the available courses for medical intervention.(Kaye et al., 1998) In the case of lung disease, the physician may better understand patient outcomes of the clinical intervention possibilities. This paper focuses

on the simulation of physically-accurate lung morphology presented in an Augmented Reality (AR) stereoscopic visualization environment. AR environments "augment" the users view of the real world with virtual information.(Tang et al., 1998) Recently these AR environments find applications from manufacturing and scientific guidance to entertainment. Visualizing 3D lung morphology in an AR environment would involve graphically super-imposing the 3D lung models onto the patient himself or a human patient simulator (HPS) and visualizing their dynamics in a physiologically-accurate manner. Fig.1 shows a snapshot of a 3D lung model super-imposed over a HPS.(Rolland et al., 2003) Such visualization may significantly improve the clinical teaching and surgical guidance of respiratory medical simulation. The key aims in achieving this visualization involved (i) tracking the real-world position of the patient or HPS, (ii) computing the subsequent global positions of the 3D lung models, (iii) registering the 3D models onto the patient or HPS and (iv) deforming and graphically rendering the 3D lung models.

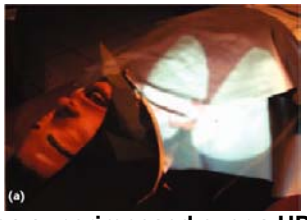


Fig 1. View of breathing lungs superimposed over a HPS in the Augmented Reality Center.(Rolland, Davis et al., 2003)

The technical challenges involved in designing a medical simulation for lung morphology and its visualization in an AR environment arise from their computational complexity. Specifically, in an AR environment the position and orientation of the patient are updated by the tracking sensor every 16 ms, which requires deforming and rendering 3D lung models at 30-66 steps per second.(Rolland, Davis et al., 2003) This subsequently limits the usage of high resolution 3D lung models for real-time deformation and visualization. A method to overcome this limitation was proposed in (Santhanam et al., 2004a). In this method the 3D lung dynamics caused by the air-flow into the lungs was modeled using Green's Formulation and the lung deformation was pre-computed and simulated in real-time for the upright position. However the changes in the 3D lung deformations for changes in the patient position and orientation in an AR environment were not accounted for. This paper extends the work in (Santhanam, Fidopiastis et al., 2004a). The method presented in this paper extends the 3D lung dynamics previously modeled in (Santhanam, Fidopiastis et al., 2004a) by off-loading the run-time deformation computations from the CPU to the GPU. The paper is organized as follows. Section 2 discusses the related work done in 3D lung morphology modeling and deformation methods implemented using programmable graphics hardware. Section 3 briefly discusses the framework developed for 3D lung morphology simulation. Section 4 discusses the proposed method for simulating 3D lung morphology and the representation

of the associated computations in the GPU. Experimental results are presented in Section 5.

2. Related Work

Lung dynamics have been investigated for the verification of thoracic medical imaging equipment and for medical training purposes.(Segars, 2002) Initial methods to model the respiratory physiology were designed as regional pressure-volume equilibrium equations based on physiology and clinical measurements. Mechanical and electrical analogues of these balancing equations were simulated for understanding the lung behavior.(Rideout, 1990) These analogues substantially aid in providing controlled mechanical ventilations.(Mesic et al., 2003)

Deformation methods compute the displacement of surface positions of a 3D virtual model. These methods have been previously employed in applications ranging from surgery simulation to animations. The deformation methods can be either geometrically-based or physically-based. The geometrical methods are mathematical methods that model the lung deformation during respiration by changing the positions of the 3D mesh surface points.(Segars et al., 2001) Such methods can accurately model the lungs change in shape in real time, but they have no physiological basis. Physically based deformation methods attempt to model the lungs with more physiological accuracy. The physically based deformation methods vary from simple functional representations such as mass-spring models,(Terzopolous et al., 1987) to representations using polynomial basis functions, such as Finite Element Methods (FEM), (Bro-nielsen, 1998) and Green's formulation.(James et al., 2003).

An FEM based deformation was proposed by DeCarlo in order to visualize 3D lung dynamics,(DeCarlo et al., 1995) and was further improved by Kaye in order to model pneumothorax related conditions.(Kaye, Primiano et al., 1998) In both of these methods a 3D lung surface model was considered as a single-compartment model.(Kaye, Primiano et al., 1998) An alternative method for modeling lung deformations using an FEM based functional representation of bronchioles and parenchyma was investigated by Tawhai.(Tawhai et al., 2003) In this method, the air-flow inside the bronchial airways was modeled using computational fluid dynamic methods. The volumetric lung space was divided into multiple compartments. The bronchial airway and the surrounding parenchyma were modeled using FEM.(Tawhai and Burrowes, 2003) The computational complexity of the FEM is limited by the number of nodes or elements used for the 3D virtual model.

Green's formulation represents the physics of non-vibrational deformation and is computationally inexpensive as compared to the FEM. This formulation has been extensively used in medical simulation of anatomical organs and their haptic

interactions.(James and Pai, 2003) Such formulation provides scope for real-time deformation of high density 3D models obtained from medical imaging methods that subsequently leads to the development of real-time medical simulation and visualization. The computational complexity of green's formulation is based on the number of elements in the 3D model used and the elastic tissue behavior. Two key mathematical components of this method are (i) the applied force on each node of the 3D model and (ii) the transfer function matrix which is a square matrix representing the tissue property. The deformation is obtained as a matrix-vector multiplcation of the transfer function matrix and the applied force vector.(Stakgold, 1979) In order to increase the computation speed, we discuss the usage of Programmable GPUs.

Programmable GPUs have been previously used for implementing both geometrically and physically-based deformation methods. The geometrically-based methods of particular importance were Key-framing and Vertex skinning.(Fernando et al., 2003) Key-framing allowed a cartoon artist animate a 3D character (model) by interpolating among a set of intermediate 3D frame sequences. Implementing this method in a GPU however had the down-side that the intermediate 3D frames needed to be stored.(Fernando and Kilgard, 2003) Vertex skinning allowed a cartoon artist to create animations by associating the movement of a node of a 3D model to the movement of set of key nodes in the 3D model. These key nodes as a group were represented as a matrix (referred to as a bone matrix) and the animation was reduced to a matrix-vector multiplication in a GPU.(Fernando, 2004) Physically-based methods were implemented in a GPU using the "Render-to-Texture" feature. This feature allowed intermediate values of a computation to be stored into a texture memory in GPU which was referred to as P-buffer.(d'Eon, 2004) This approach was used in obtaining physically-based wave motions for fluid and cloud simulations, (Harris, 2004) and FEM computations (Rumpf et al., 2001) in a GPU.

The focus of our research is on obtaining real-time deformations of high-resolution 3D lung models and to visualize them in an AR environment. We off-load the deformation's computation from the CPU to the GPU. Specifically while the CPU caters to the computational requirements of AR environment the GPU caters to the computational requirements of deforming and rendering the 3D model in this AR environment.

3. Proposed Method for 3D Lung Deformation

In this section we outline the methodology adopted for the dynamic simulation of 3D lungs deformation. This method is sub-divided into three stages of which the first two stages have been previously addressed in (Santhanam et al., 2004c) and (Santhanam, Fidopiastis et al., 2004a). In the first stage we parameterize the change in lung volume for a change in pressure (referred to as trans-pulmonary pressure).(Santhanam et al., 2004b; Santhanam, Fidopiastis et al., 2004c) This change in pressure which caused the air-flow inside lungs accounts for both the local muscle resistance caused by anatomical

components such as lung tissue, diaphragm and rib-cages as well as the motor drive of breathing controlled by a network of neurons in the medulla. This relation between the lung volume and the trans-pulmonary pressure was referred to as a Pressure-Volume (PV) relation. Both normal and abnormal PV relations may be simulated.

In the second stage we estimated the change in the global lung shape for an increase in lung volume.(Santhanam, Fidopiastis et al., 2004a) This is obtained using a physically based deformation method. Within the context of computer animation, a green's function based deformation was chosen since it has been observed that lung deformations do not undergo vibrations.(Mead, 1956) The total number of nodes on each of the 3D high-resolution lung models is approximately 400,000 which is high as compared to the usage of 500 elements in (Kaye, Primiano et al., 1998). Such a large number of nodes facilitate effective modeling of both normal and patho-physical lung deformations. Also a Young's modulus was first associated to every node of the 3D lung model based on the lung's regional alveolar expansion. A unit force was then applied on each node and the transfer function matrix was computed using an iterative approach. In each step of the iteration the force applied on a node was shared with its neighboring nodes based on a local average of Young's modulus as described in (Santhanam, Fidopiastis et al., 2004a). The iteration stopped when this sharing of applied force reached equilibrium. At this point of equilibrium the force shared by a node with its neighbors formed a row of the transfer function matrix. In order to deform the 3D lung model an upright orientation was considered. The applied force due to the air-flow inside lungs was given by the vertical pressure gradient of lungs.(West, 1995; Fan et al., 2001; Krishnan et al., 2004) The computed force was then normalized so that the sum of applied force magnitude on all the nodes was equal to a unit increase in volume. A unit increase in volume was set as the ratio between the tidal volume of human lungs (i.e, 500 ml) and the product of the deformation steps per second (i.e, 66.66 steps/sec) multiplied by the ventilation rate of inhalation or exhalation (normally 3 sec/breathing). Thus for an increase in lung volume the subsequent change in lung shape was computed.(Santhanam, Fidopiastis et al., 2004a)

The third stage of the proposed methodology forms the focus of this paper. We present a method to optimally compute the matrix-vector multiplication in a GPU during run-time. Specifically the matrix-vector multiplication is represented in steps which can be pre-computed off-line partially and is explained as follows. The columns of the transfer function matrix are pre-computed and represented using spherical harmonic (SH) coefficients. These coefficients are obtained from orthonormal decomposition of the transfer function matrix using SH transformations.(MacRobert et al., 1967) A unique property of lung deformations we make use in our approach is the property that the transfer function remains a constant until the tissue properties undergo irreversible damage.(Hauth et al., 2003) Since lungs do not undergo irreversible damages under normal breathing conditions, the transfer matrix can be considered a constant. The transfer function matrix is pre-computed using the method proposed in (Santhanam, Fidopiastis et al., 2004a) and its SH coefficients are transferred into the local memory of the GPU along

with the 3D model before starting the simulation. The SH coefficients of the force applied on the 3D model are also pre-computed. During the simulation, 3D deformation is computed in GPU as a dot product of pre-computed SH coefficients of the applied force and the transfer function matrix, which is computationally inexpensive when compared to the dot product of the transfer function matrix and the applied force for a high-resolution 3D model. The usage of this representation in deforming the 3D lung model is as explained in the section 4.

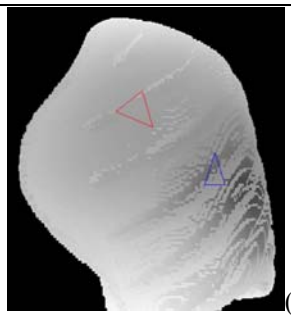
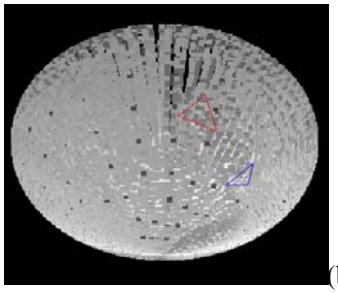

(a)

(b)

Fig 2a-b. Fig.2a shows the 3D polygonal lung model and Fig 2b shows the polar coordinate representation.

4. Mathematical Model

In this section we discuss the steps in the third stage of the methodology adopted for simulating 3D lung dynamics. The nodes of the 3D lung model (as shown in Fig.2a) are represented using polar coordinates (as shown in Fig.2b). Let θ_I, ϑ_I represent the polar coordinate of node I of the 3D lung model. At any instant during deformation a force is applied on each node. Based on the values of the force applied on each node at that instant we can determine the function $F(\theta_J, \vartheta_J)$ that best approximates them.(Braess, 1986) Similarly we can also determine the function $T(\theta_I, \vartheta_I, \theta_J, \vartheta_J)$ that best approximates the transfer function matrix of the 3D lung model which represents the elastic interaction between node *I* and *J* associated with the deformation. The fundamental form of the transfer matrix is as given in (MacRobert and Murray, 1967; Stakgold, 1979). Let $D(\theta_I, \vartheta_I)$ be the displacement of the node I. For simplicity in notation we denote $D(\theta_I, \vartheta_I)$ by *D(I)*, $F(\theta_J, \vartheta_J)$ by *F(J)* and $T(\theta_I, \vartheta_I, \theta_J, \vartheta_J)$ by *T(I,J)*. *D(I)* using Green's formulation can be written as,

$$D(I) = \sum_{J=0}^{N} F(J) \cdot T(I,J) \ , \qquad (1)$$

where *N* is the total number of nodes in the 3D lung model. Further to reduce the complexity of the computation we approximate this function using a set of SH basis

functions (polynomials). The displacement and the transfer matrix have been previously shown to be elements of a Hilbert space which can thus be re-represented using SH polynomials.(Reddy, 1998) We use this theoretical fact to modify the computations involved in the dot product on the right hand side of equation (1). A finite set of SH coefficients are used for representing an array of numbers. Let $Y_{lm}(\theta_J, \vartheta_J)$ represent the SH polynomials and $Y^*_{lm}(\theta_J, \vartheta_J)$ be their conjugates polynomial at node *J*.(Ramamoorthi et al., 2001) For simplicity in notation we denote $Y_k(\theta_J, \vartheta_J)$ by $Y_k(J)$ and $Y^*_k(\theta_J, \vartheta_J) \bullet \sin(\theta_J)$ by $Y^*_k(J)$. The applied force can be represented as

$$F(J) = \sum_{l=0}^{n\_bands} \sum_{m=-l}^{l} F_{lm} \cdot Y_{lm}(J) , \qquad (2)$$

where *n_bands* specifies the total number of SH bands and is the square root of the total number of SH coefficients. The values of F_{lm} are the SH coefficients of the force where *l* and *m* are indexes for the SH coefficients and the SH polynomials. The SH coefficients can be computed from the applied force by the following relation

$$F_{lm} = \sum_{J=0}^{N} F(J) \cdot Y*_{lm}(J) . \qquad (3)$$

The I^{th} row of the transfer function can now be given as

$$T(I,J) = \sum_{l=0}^{n\_bands} \sum_{m=-l}^{l} T^I{}_{lm} . Y_{lm}(J) , \qquad (4)$$

where $T^I{}_{lm}$ represents the SH coefficients of the I^{th} row of transfer matrix. Now we re-write equation (1) by substituting the expansions of the applied force and the transfer function row with their expansions given in equation (2) and (4) respectively. Thus equation (1) can be written as

$$D(I) = \sum_{J=0}^{N} \left[\sum_{l=0}^{n\_bands} \sum_{m=-l}^{l} F_{lm} . Y_{lm}(J) \right] \cdot \left[\sum_{l'=0}^{n\_bands} \sum_{m'=-l'}^{l'} T^I{}_{l'm'} . Y_{l'm'}(J) \right] . \qquad (5)$$

It is worth mentioning the orthonormal property of SH functions. The summation (for values of J from 0 to N) of the product of Y_{lm} and $Y_{l'm'}$ is 1 when *l* equals *l'* and *m* equals *m'* are equal and 0 otherwise.(Reddy, 1998) We thus reduce the equation (6) to be represented in terms of the SH projected function

$$D(I) = \sum_{l=0}^{n\_bands} \sum_{m=-l}^{l} F_{lm} \cdot T^I{}_{lm} \cdot A , \qquad (6)$$

where *A* is a constant (of value 1) representing the summation of the square of Y_k (whose value is 1 due to its orthonormal property) for all values of *J*.(Reddy, 1998) We now introduce the following modification to equation 6.

$$D(I) = \alpha \sum_{l=0}^{n\_bands} \sum_{m=-l}^{l} (F_{lm} \cdot T^{I}{}_{lm} \cdot A), \quad (7)$$

where α is a coefficient which allows to account for finite value of *n_bands*. Further discussion on the choice of n_bands and the corresponding value of α are discussed in Section 5. The SH coefficients of both the applied force and the transfer function matrix can be pre-computed and thus we need to compute only equation (7) during run-time in order to obtain deformation of the 3D model.

Having discussed a method to optimize the dot product, we propose to compute SH coefficients of the applied force for different patient's orientation in run-time by interpolating among a set of specified orientations. The origin of the coordinate system is set to be the centriod of the lung model. The orientation of the lung model is now represented in terms of the rotation angles along the *X, Y* and *Z* axes. Let p^i,q^i,and r^i be the arrays of pre-computed SH coefficients for the applied force at rotation angle *i* of range in the set $\pi/2,\pi,3\pi/2$, and 2π along *X, Y,* and *Z* axes respectively. Let *a, b* and *c* be arbitrary values for current rotation angles for the 3D lung model. For such orientation, the SH coefficients for the applied force are computed by smoothly interpolating among p^i,q^i,and r^i. Let a_0, a1, *b0, b1, c0* and *c1* be the angles of *i* that form the closest lower and upper limits *for a,b,* and c, respectively. The SH coefficients of the applied force for the orientation *a, b*, and *c* are now given by

$$f^{abc}{}_{lm} = f^{a}{}_{lm} + f^{b}{}_{lm} + f^{c}{}_{lm} \quad (8)$$

where

$$f^{a}{}_{lm} = \frac{\left[p_{lm}{}^{a0} \times \cos(a-a0) + p_{lm}{}^{a1} \times \cos(a-a1)\right]}{3} \quad f^{b}{}_{lm} = \frac{\left[q_{lm}{}^{b0} \times \cos(b-b0) + q_{lm}{}^{b1} \times \cos(b-b1)\right]}{3},$$

$$f^{c}{}_{lm} = \frac{\left[r_{lm}{}^{c0} \times \cos(c-c0) + r_{lm}{}^{c1} \times \cos(c-c1)\right]}{3}. \quad (9)$$

The advantage of using SH coefficients is that it reduces the dot product of two arrays of variable length of non-zero values to a dot product of two arrays of fixed length of non-zero values. Since the SH coefficients of the transfer function matrix as well as the applied force are pre-computed, a significant amount of computation time is saved.

We now summarize the third stage in three steps.

1. In the first step, the position, color and normal of a node in the 3D lung model as well as the transfer function matrix are transferred into the GPU's vertex array. Since GPU is a per-vertex processor, every row of the transfer matrix is associated with a node of the 3D model and is transferred into the vertex array as multi-texture coordinates.
2. The second step involves computing the SH coefficients of the applied force array using equation (8-9) for the current orientation of lungs. The AR steps in

tracking the orientation of the patient or HPS is beyond the scope of this paper and thus the orientation angles are assumed to be obtained from AR components.

3. The third step of the third phase is the implementation of equation (7) in GPU at every frame. We use a GeForce4 FX Ti4200 for rendering and deforming the 3D lung model. Specifically we use a CG 1.1 based programmable vertex-shader,(Fernando and Kilgard, 2003) for the shading of the 3D model and implementing the dot-product of the per-vertex transfer matrix's row and the applied force.(Fernando and Kilgard, 2003)

We now investigate the mathematical computational requirements of equation (1). For High-Resolution 3D lung models, it was observed that each row of the transfer function matrix has an average of 50 non-zero values. Thus the computation of deformation of a node, using equation (1) will undergo an average of 50 multiplications and 49 additions. We now continue by analyzing the computational requirements of equation (7). We first compute the SH coefficients of the transfer function, which can be performed off-line as the transfer function for lungs is considered a constant. The SH coefficients of the applied force at pre-determined orientations can be computed from equation (3) offline. For any orientation of the 3D lung model the applied force is interpolated using equations (8-9) The deformation computation of equation (7) would be 16 multiplications and 15 additions per node which reduces the number of computations by approximately one-third.

5. Lung Deformation Results

In this section we first discuss the speed-up obtained in using GPU-based deformations. It is followed by a discussion on the graphic outputs obtained from those deformations. Fig 3a presents a comparison of the lung ventilation (volume change with time) visualized using the equation (1) for the deformation in a CPU (2800+ AMD Athlon) with the ventilation visualized using equation (7) in a GPU (NVIDIA GeForce4 Ti4200). It can be seen that the difference (referred as Lag) between the lung's ventilation using a GPU-based deformation (i.e. blue line in Fig.3a) and the actual ventilation simulated using only the first phase of the proposed framework (i.e. red line in Fig.3a) is negligible. Such small difference can be explained from the off-loading of the deformation computations from the CPU to the GPU. Thus a 3D high-resolution lung model deformation is obtained with a negligible lag in simulation. A lag in simulation is however observed in the case of a CPU-based high-resolution 3D lung deformation (i.e. pink line in Fig.3a). This lag in CPU based deformation increases linearly with simulation time as shown in Fig 3b. These graphs support the computational speed-up results obtained from using GPU.

A snapshot of the visualization of high-resolution 3D lungs in the upright position at the start of an inhalation phase is shown in Fig4a. The change in shape and volume at the end of the inhalation phase using a GPU based computation is shown in Fig 4b. The difference in this change in shape and volume was negligible when compared to the change in shape and volume obtained using the CPU based computation. A side-view of

lung dynamics in upright and supine positions are shown in Fig 5a and 5b respectively. The subtle difference in the deformation of the base of the lungs in the upright and supine positions can be seen. Specifically the front side of the base region in the supine position displaces more when compared to the back side of the base region in the supine position which is not observed in the upright position. The total number of SH coefficients used in the GPU-based implementation is 16, as it yield α close to 1 in using equation (7) for lung deformations. The value of α was experimentally found to be 1.03. For less than 16 coefficients the value of α dropped significantlty below 1. For values more than 16 the value of α stayed around 1. An accurate number of SH coefficients and the value of α needed for modeling patho-physical lung deformations need to be established in future work.

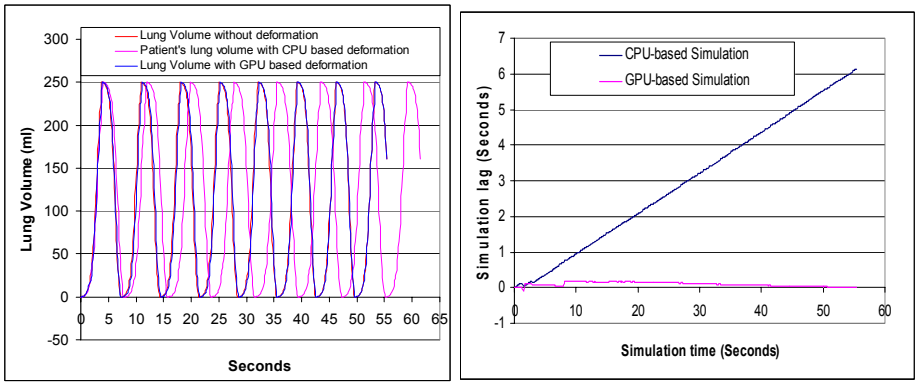


Fig 2 (a) Delay caused by the usage of CPU based lung deformation and the subsequent optimization seen in a GPU based lung deformation. (b) Demonstraton of the increase in simulation lag with the usage of the CPU based deformations as opposed to GPU.

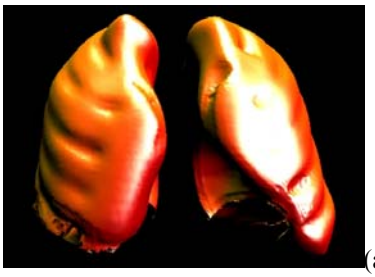
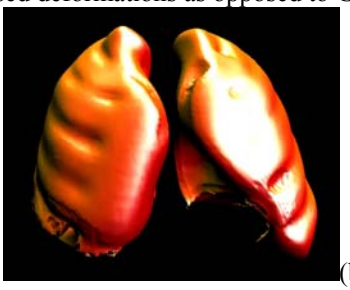

(a)

(b)

Fig 3 . 3D lung shape (frontal view) in an upright position at (a) the start of inhalation and (b) at the end of inhalation.

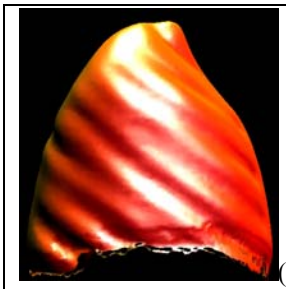
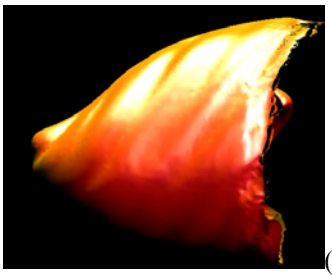
 (a)  (b)

Fig 4 . 3D lung shape (side view) at the end of inhalation (a) in an upright position.and (b) in a supine position.

6 Conclusion

In this paper we have discussed a method to compute 3D lung deformations in a GPU for an AR environment that accounts for changes in the deformations for changes in the orientation of the patient or HPS. Such visualization may play a significant role in assessing clinical interventions for a patient. The usage of high-resolution models in the visualization supports meticulous modeling of tissue-degenerations. Future work will involve validating the number of SH coefficients used and applying the proposed framework for visualization of patho-physical lung morphology.

References

Braess, D., *Nonlinear Approximation Theory*, Springer Verlag.(Berlin, 1986).

Bro-nielsen, M., Finite Element Modeling in Medical VR, 1998, Journal of the IEEE 86, (3), 490-503.

DeCarlo, D., J. M. Kaye, D. Metaxas, J. R. Clarke, B. Webber and N. I. Badler, Integrating anatomy and physiology for behavior modeling, 1995. Medical Meets Virtual Reality 3, San Diego

d'Eon, E., Deformers. GPU Gems, 2004, Fernando,R., Pearson Education.

Fan, L., C.-W. Chen, E. A. Hoffman and J. M. Reinhardt, Evaluation and application of 3D lung warping and registration model using HRCT images, 2001. Proc. SPIE Conf. Medical Imaging, San Diego,CA234-243.

Fernando, R., *GPU Gems*, Pearson Education.(Boston,MA, 2004).

Fernando, R. and M. Kilgard, *The CG tutorial*, Pearson Education.(Boston,MA, 2003).

Harris, J., Fast fluid dynamics simulation of GPU. GPU Gems, 2004, Boston,MA, Pearson Education 637-665.

Hauth, M., J. Gross, W. Straber and G. F. Buess, Soft-tissue simulation based on measured data, 2003, Medical Image Computing and Computer Aided Intervention 262-270.

James, D. L. and G. K. Pai, Multiresolution Green's Function Methods for interactive simulation of elastostatic models, 2003, ACM Transactions of Graphics 22, (1), 47-82.

Kaye, J. M., F. P. J. Primiano and D. N. Metaxas, A Three-dimensional virtual environment for modeling mechanical cardiopulmonary interactions, 1998, Medical Image Analysis 2, (2), 169-195.

Krishnan, S., K. C. Beck, J. M. Reinhardt, K. A. Carlson, B. A. Simon, R. K. Albert and E. A. Hoffman, Regional lung ventilation from volumetric CT scans using image warping functions, 2004.

MacRobert and T. Murray, *Spherical Harmonics an elementary treatise on harmonic functions with applications*, Oxford Pergamon Press.(New York, 1967).

Mead, J., Measurement of Inertia of the Lungs at Increased Ambient Pressure, 1956, Journal of Applied Physiology 2, (1), 208-212.

Mesic, S., R. Babuska, H. C. Hoogsteden and A. F. M. Verbraak, Computer-Controlled Mechanical Simulation of the Artificially Ventillated Human Respiratory System, 2003, IEEE Transactions on Biomedical Engineering 50, (6), 731-743.

Ramamoorthi, R. and P. Hanrahan, On the relationship between radiance and irradiance: determining the illumination from the images of complex lambertian object, 2001, Journal of Optical Society of America A, 2448-2459.

Reddy, D., *Introductory Functional Analysis with Applications to Boundary Value Problems and Finite Systems*, Springer Verlag.(New York, 1998).

Rideout, M., *Mathematical modeling of Respiratory systems*, Printice Hall.(New Jersey, 1990).

Rolland, J. P., L. Davis and F. Hamza-Lup, Development of a training tool for endotracheal intubation: Distributed Augmented Reality, 2003, Medicine Meets Virtual Reality (MMVR) 11, 288-294.

Rumpf, R. and R. Strzodka, Using Graphics cards for Quantized FEM Computations, 2001. Proceedings of IASTED Visualization, Imaging and Image Processing Conference193-202.

Santhanam, A., C. Fidopiastis, F. Hamza-Lup, J. P. Rolland and C. Imielinska, Physically-based Deformation of High-resolution 3D lung models for Augmented Reality based Medical Visualization, 2004a. Medical Image Computing and Computer Aided Intervention, AMI-ARCS, Rennes,St-Malo, Lecture Notes on Computer Science,21-32.

Santhanam, A., C. Fidopiastis and J. P. Rolland, An adaptive driver and real-time deformation algorithm for visualization of high-density lung models, 2004b. Medical Meets Virtual Reality 12, Newport, CA, IOS Press,333-339.

Santhanam, A., C. Fidopiastis, J. P. Rolland and P. Davenport, A bio-mathematical formulation for modeling the pressure-volume relationship of lungs (in Press), 2004c, IEEE Transactions on Information Technology and Bioinformatics.

Segars, W. P., Development of a new dynamic NURBS-based cardiac-torso (NCAT) phantom, (2002). Chapel-hill, University of North Carolina.

Segars, W. P., D. S. Lalush and B. M. W. Tsui, Modeling Respiratory mechanics in the MCAT and the spline-based MCAT systems, 2001, IEEE Transactions on Nuclear Science 48, (1), 89-97.

Stakgold, I., *Green's functions and boundary value problems*, Wiley Interscience1979).

Tang, S.-L., C.-K. Kwoh, M.-Y. Teo, N. W. Sing and K.-V. Ling, Augmented reality systems for medical applications, (1998). Engineering in Medicine and Biology Magazine. 17, 49-58.

Tawhai, M. H. and K. S. Burrowes, Developing integrative computational models, 2003, Anatomical Record 275B, 207-218.

Terzopolous, D., J. Platt, A. Barr and K. Fleisch, Elastically Deformable Models, 1987, ACM Siggraph 1987 21, (4), 205-214.

West, J. B., *Respiratory Physiology, the essentials*, Lippincott Williams and Wilkins.(Philadelphia,USA, 1995).