A Multi-GPU Framework for Simulating High-Resolution Bio-mechanical Head and Neck Deformations

John NEYLON[a], Patrick KUPELIAN[1], and Anand SANTHANAM[1]

[1] *Department of Radiation Oncology, University of California, Los Angeles*

Abstract. The aim of this paper is to enable a multi-GPU computational framework for simulating high-resolution biomechanical head and neck deformations for radiotherapy dose monitoring purposes. The biomechanical model incorporates subject-specific young's modulus and shear modulus properties and is actuated using daily patient positioning images. The computational tasks of the biomechanical model were off-loaded to a multi-GPU framework enabling real-time biomechanical deformations. The results show the biomechanical head and neck model could be deformed at a rate of 20-40 frames per second. The model deformations indicated changes in the stress at different parts of the treatment as the head and neck posture varied from one treatment fraction to another.

1. Introduction

Advances in radiotherapy treatments have significantly improved the patient's quality of life by enabling clinicians to effectively treat a target anatomical region. Of particular importance are the advancements in the image-guidance where one or more 3D or 4D images at multiple resolutions are used for positioning the patient during a clinical intervention. Our *ultimate goal* is to further improve image-guided clinical interventional efficacy by largely eliminating normal tissue damage and the other treatment side effects. Such a goal would require the development of biomechanical models that can actuate (e.g. move, wiggle, twist etc) to represent the patient's position when placed on the couch. Incorporating such biomechanical models of the subject's targeted anatomy will be necessary for predicting the dose to be delivered. [1,2]. While Computed Tomography (CT) imaging provides complete patient anatomy for a given posture; on-board mega-voltage CT imaging [3,4] provides a grainy anatomical representation of the region of interest. Undetected and uncompensated registration errors between the CT and MVCT can lead to suboptimal prediction of the dose to be delivered. With the current standard of care, the therapist first positions the patient in the imaging couch using positioning lasers. Once positioned, the patient's posture is adjusted by registering only the bone structures observed in the MVCT. However, for a perfect posture, both the soft tissue as well as the bone anatomy needs to be taken into account. In cases where the tumor is in the middle of critical organs (e.g. Parotid glands), patient discomfort may cause the anatomy to move or deform causing mis-calculations in the delivered dose. An image registration process that accounts

for such voluntary and involuntary changes in the patient's gross anatomy and posture is required for enabling precise patient positioning and prediction of the delivered dose. Our focus is on developing a biomechanical head and neck model that forms a first step towards model guided image registration of CT and MVCT for the head and neck anatomy. Such biomechanical models have been previously developed and investigated for applications ranging from animations [5] to medical image registration [6,10] and treatment simulations [7,8]. For model guided image registration purposes, we envision the biomechanical model to have a resolution of 1mm-3mm thereby increasing the model elements to be of the order of 10^6. The computational complexity of such models limits them from real-time deformations. The focus of this paper is to present a multiple Graphics Processing Units (GPUs) system for real-time biomechanical head and neck model deformations.

2. Materials and Methods

This section is further sub-divided as follows: Section 2.1 discusses the patient data collection and the model generation steps. Section 2.2 presents the steps taken towards the multi-GPU interprocess communication for deformation purposes. Section 2.3 discusses the methods used to account for deformation discontinuities caused by the multi-GPU usage.

2.1 Patient data acquisition

Our software read in a patient CT, fills the structure while differentiating between air, bone, and soft tissue using a simple threshold method. A single mass element was added to the system for each voxel and positioned within the system world according to voxel address.

The user also had the option of loading an RTSTRUCT format file, and choosing specific contours, which allowed the user to assign and adjust characteristic parameters for each individual contour separately from the generic soft tissue of the model.
A mass-spring model was designed for deforming the head and neck anatomy. During initialization the mass-spring connections were established through a grid-hash sorting technique. This divided the system world into a uniform cell grid, and assigns every mass element a hash value identifying its containment cell. Mass elements were then sorted by cell id using a fast radix algorithm. This then allowed a search through neighboring cells for mass elements. Currently the search area is a 5x5x5 cell region. If the mass elements were within a specified distance, the vector between the mass elements was recorded as the rest length of the spring and rest state orientation. A second array was created which held the mass elements id of every connection for each mass elements.

After initialization, the system entered its update loop, constantly refreshing the 3D interactive simulation. The user had the option of setting the time step, which artificially quantifies the amount of time that passes in the simulation universe between frame updates, to the inverse of the frame rate and running the simulation in real time.

The update loop incorporated all physics and constraints, using the previous system state, to calculate the new positions of every mass elements. This involves several kernels, calculating force and response before integrating the final position.

After initially encountering issues during CT reconstruction of our deformed data sets, a second stage was added to the integration method, splitting the internal and external interactions of the currently controlled contour. The algorithm begin by calculating the force due to changes in the rest length of the currently controlled contour. As mentioned previously, the user was able to load a structure file, and select which contours they would like to control separately. They can then cycle through these contours, individually adjusting the rest length. For example, if the rest length of the PTV was reduced, the first stage launched a CUDA kernel which calculated the composite force on each mass elements within the PTV by looping over each of its connections and calculating the force from only the connected mass elements that are also within the PTV.

2.2 Off-loading calculations to a multi-GPU framework

Modeling the complex biomechanical head and neck motion for a high-definition anatomy is computationally very expensive. At each iteration, the computational complexity was O(N log M), where N is the number of voxels representing the anatomy (~ 1 M voxels) and M is the local neighborhood of each of the voxel (27 M springs). Deforming such a huge model forms using a multi-GPU framework forms the focus of this section. We used a multi-process model for achieving real-time computations. Specifically, each GPU was allocated with a server process that receives and performs computational tasks. A shared memory was allocated in the system in order for each of the GPU processes to access the model data. To enable a serialized data access, we used a semaphore-based control for the shared memory. The central client process copied data into the shared memory and formulates a parallel access to the different sections of the biomechanical model. Each GPU process also enabled an overlapping region in the biomechanical model for smoother deformation integration. The model deformation by each of the GPU process was computed as discussed in section 2.1. The deformation vectors for each of the GPU processes were then integrated, after each iteration, for structures that overlap between multiple GPUs. Finally, a dedicated GPU was used for rendering the deformed biomechanical model.

2.3 Accounting for deformation discontinuity in a multi-processor computing setup

The complex biomechanical head and neck deformations led to deformation discontinuities caused by the anatomical barriers imposed by each GPU. To enable a multi-GPU deformation mechanism, we divided the contour-filled head and neck anatomy into sections that can be allocated for each of the GPU at the start of the head and neck simulation. Specifically, the 3D mesh representing the anatomy was divided into sub-sections in such a

way that each contoured 3D structure comes under a single section. To this end, we employed a CVGA clustering process using the MacQueen's algorithm.
The steps involved in this algorithm were as follows:

a. An initial sectioning was performed based on the contour-filled muscle structures
b. New sections were obtained by switching a mass element from one cluster to another. MacQueen's method randomly picks a set of *k* mass elements in a contoured muscle structure and added to a nearby cluster representing a different muscle structure. The "closeness" of the mass element to the muscle structure was determined by using the mean distance of mass elements that belong to that contoured structure.
c. Each mass element was updated to recalculate the distance to the updated muscle structure. If the closest muscle structure for the point is not the one it currently belongs, the point will switch to the new section. When switching occurs, centroids of both modified clusters were updated.
d. The steps are repeated until all the mass elements are updated at least once.
e. The sections are finally grouped into the smaller groups equal to the number of available GPUs and each section is then allocated to the corresponding GPU.

3. Results

Fig 1a-c illustrates the multi-GPU allocation of the critical structures for deformation purposes. In this case, the PTV along with the parotids were allocated to a GPU (Fig 1a) while the neck muscles and the post avoid region were allocated to two other GPUs, respectively (Fig 1b and 1c). From a computational perspective, it can be seen that the GPU allocation process is very complex *(O(n.m)),* where *n* is the number of voxels and *m* is the number of voxels used for the cluster verification. However, the GPU allocation is performance only once during the model development stage and so can be effectively pre-computed for repetitive model usage.

The results of biomechanical head and neck simulations using a multi-GPU framework are **now discussed. Fig 2a shows the rest state of the biomechanical model. The elements** representing CT voxels are color-coded representing tissue contraction (green through blue) and stretching (green through red). Head and neck rotations are simulated by rigidly rotating the skeletal structure. Fig 2b and 2c shows the biomechanical deformation caused by skull and neck discs rotation along the body axis (head to toe). The differences in the deformation shows the subtle muscle deformations caused by changes in the patient posture during radiotherapy treatment.

Variations in the deformations for radiation critical structures in the head and neck region are presented in fig 3. Specifically, fig 3a shows the rest state of a biomechanical model consisting of the head and neck skeletal structure and a primary target volume

(PTV)- the anatomical region to be treated using radiothereapy, parotid glands and neck muscles. Other muscle structures are excluded for this simulation. Fig 3b-c shows the biomechanical deformation of the critical structures caused by the skull and neck discs rotation along the body axis. Fig 3c shows the deformations in the head and neck region caused by just the skull. The differences in the deformation shows the subtle muscle deformations caused by changes in the patient posture during radiotherapy treatment. Fig 4a-c shows the posterior neck muscle deformation when the skull is actuated to assume different postures. The strain energy on each of the voxel is color-coded (blue, green, yellow and red) representing the strain energy when the head and neck region undergoes different postures.

The computational speed of the multi-GPU head and neck deformation system was examined using two systems A and B, respectively. System A consists of 2 Nvidia GTX 680m and system B consists of 2 Nvidia GTX 680. Using multi-GPU we obtained frame rates of the order of 16 Frames per second using system A and 20 FPS for system B. When only the region of interest was included (Fig 2 and 3), we obtained a performance of 40 FPS for system A and 60 FPS for system B. No specific speedup in multi-GPU was observed since the calculation fitted into a single GPU. Nevertheless, the inter-GPU communication added a communication overhead, which slowed the multi-GPU implementation when compared to a single GPU implementation.

Conclusion

In this paper, we presented a multi-GPU biomechanical head and neck deformation model for radiotherapy purposes. A semaphore controlled shared memory interface enables inter-GPU communication. Our results showed that highly complex biomechanical model deformations can be obtained at real-time FPS. Simplifying the model by including only the region of interest further enables an improved run time but introduces calculation differences when compared to the full anatomy scenario. A comprehensive biomechanical model will thus include all the anatomical structures and their accurate biomechanical model in order to obtain a quantitatively accuracy deformation.

Acknowledgement

This work is funded by the University of California, Los Angeles.

References

1. Wanatabe, H., et al., “Swept source optical coherence tomography as a tool for real time visualization and localization of electrodes used in electrophysiological studies of brain in vivo”, Biomedical Optics Express 2(11), 3129-3134 (2012).
2. Eckstein, C., et al., “Detection of clinical and subclinical retinal abnormalities in neurosarcoidosis with optical coherence tomography”, Journal of Neurology 1, 63-68 (2012).
3. Rolland, J.P., P. Meemon, S. Murali, K. P. Thompson, and K. S. Lee, “Gabor-based fusion technique for Optical Coherence Microscopy”, Opt. Express 18(4), 3632-3642 (2010).
4. Lee, K.S., K.P. Thompson, P. Meemon, and J.P. Rolland, “Cellular resolution optical coherence microscopy with high acquisition speed for in-vivo human skin volumetric imaging”, Optics Letters 36(12), 2221-2223 (2011).
5. Lee, S.H., et al., Comprehensive biomechanical modeling and simulation of the upper body. ACM Siggraph, vol 99 1-17 (2010).
6. Santhanam, A., Y. Min, S. Mudur, E. Divo, A. Kassab, B. H. Ruddy, J. Rolland, P. Kupelian. 2010 (In press) A Hyper Spherical Harmonic Formulation for reconstructing volumetric 3D lung deformations. Comptes Rendus Mechanique “Special Issue on Inverse Problems”
7. Santhanam, A.P., T. Willoughby, S.L.Meeks, and P. Kupelian. 2009. Modeling simulation and visualization of 3D lung conformal dosimetry. Physics in Medicine and Biology 54 6165-6180.
8. Min, Y., A. Santhanam, A., Y. Min, H. Neelakkantan, B.H. Ruddy, S. Meeks, P. Kupelian. 2010 (In press) A GPU based framework for modeling real-time 3D lung tumor conformal dosimetry with subject-specific lung tumor motion. Physics in Medicine and Biology.
9. Barber, J.R., Elasticity 3rd edition. (2009).
10. Santhanam, A., et al, A Multi-camera based interfraction and intrafraction tracking system for head and neck. Journal of Medical Physics (submitted) (2012).
11. Santhanam, A., et al., Biomechanical head and neck modeling for interfraction and intrafraction tracking. American Association of Physicists in Medicine Annual Meeting (submitted) (2012)

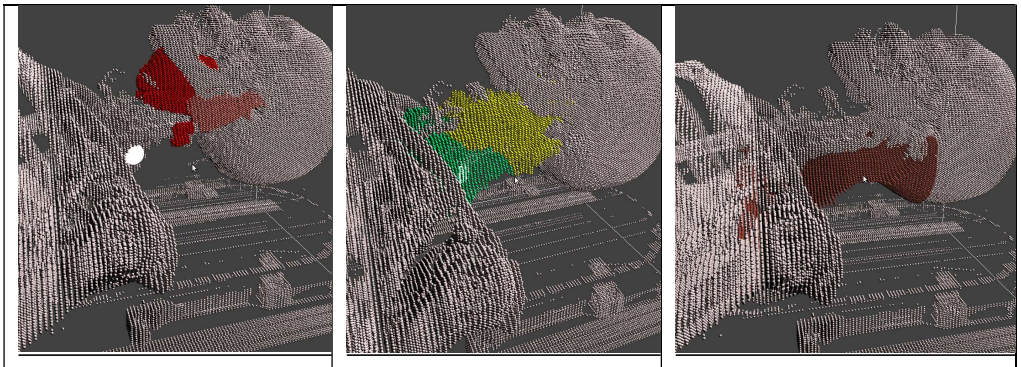

Fig 1 Allocation of different structured contours to GPUs. The skeletal positions are used as rigid body constraints by all the GPUs. The contoured regions are allocated to each GPU based on the CVGA analysis. In the case of a 3 GPU allocation, the PTV(red) and the parotids are allocated to a single GPU (a). The neck muscles were allocated to another GPU (b). The post-avoid region of the neck was allocated to another GPU (c).

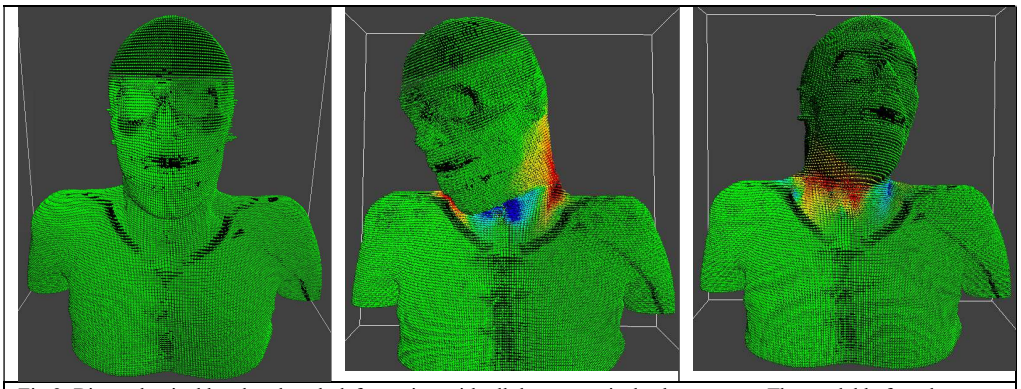

Fig 2. Biomechanical head and neck deformation with all the anatomical substructures. The model before the deformation is shown in (a). Two different neck rotations are demonstrated in (b) and (c).

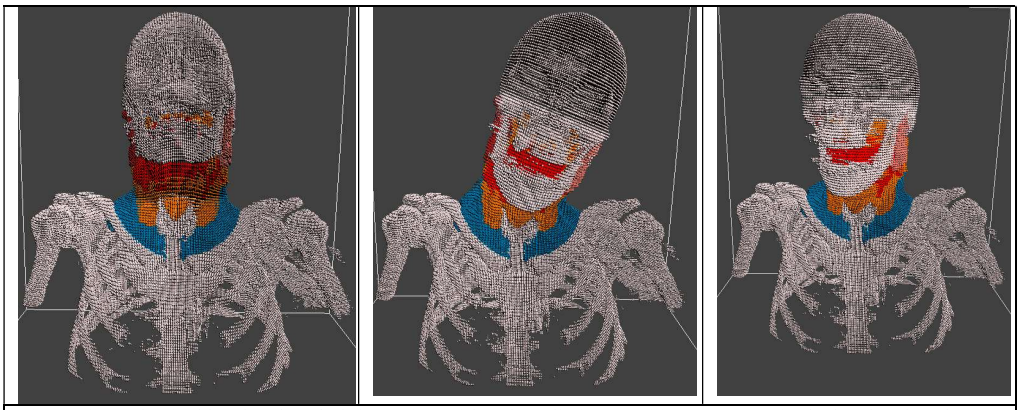

Fig 3. Biomechanical head and neck deformation with only radiation sensitive structures in the head and neck region. The model before the deformation is shown in (a). Two different head and neck rotations are demonstrated in (b) and (c).

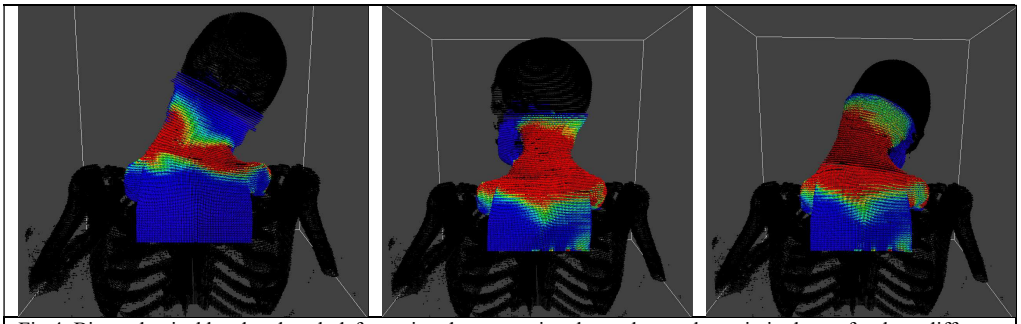

Fig 4. Biomechanical head and neck deformation demonstrating the neck muscle strain is shown for three different neck rotations.