We present a biomechanical model based four-dimensional computed tomography (4DCT) simulation method for examining the patient lung deformation induced by respiratory motion, given only one CT scan as an input. First, we model the lung stress-strain behavior using a sophisticated hyperelastic model, and solve the lung deformation problem through finite element (FE) analysis. We introduce robust algorithms to segment out the diaphragm control points and spine regions to carefully define the boundary conditions and loads, and to improve the FE convergence through our mesh optimization algorithm. Next, we treat the remaining CT volume as discretized mass points connected by springs and dampers, and simulate the motion of liver, bones, and other organs using finite difference analysis. This novel heterogeneous design can leverage the advantages of both continuum mechanics and mass-spring-damper system in the way that the lung deformation is computed in very high accuracy while the deformation of the rest CT volume can be achieved under practical computation constraints. Experimental results through comparing with the manually labeled landmark points in real patient 4DCT data demonstrate that our 4DCT simulator is very accurate.

Keywords: 4DCT, biomechanics, medical image processing, finite element analysis, radiation therapy, mesh models, respiratory motion, lung deformation.

1. Introduction

The use of four-dimensional computed tomography (4DCT) has becoming a common practice in radiation therapy, especially for treating tumors in thoracic areas. The information from 4DCT scan allows the radiation oncologist to design more accurate treatments for moving tumors, better target these moving tumors and deliver radiation within a certain interval in the breathing cycle, and reduce the risk of treatment-related side effects.

There are two alternative methods for 4DCT acquisition, namely retrospective slice sorting and prospective sinogram selection. For the retrospective slice sorting method, the projection data are continuously acquired for a time interval longer than a full respiratory cycle, multiple slices corresponding to different acquisition time points are reconstructed and then sorted into respiratory phase bins using various respiratory signals. For prospective sinogram selection method, the scanner is triggered by the respiratory signal. Then, the projection data within the same phase bin are used to reconstruct CT slices corresponding to that breathing phase.

No matter which method is used, the prolonged acquisition time results in a considerably increased radiation dose. For example, the radiation dose of a standard 4DCT scan is about 6 times of that of a typical helical CT scan and 500 times of a chest X-ray. Moreover, 4DCT acquisition cannot be applied to determine the tumor position in-situ. These facts have become a major concern in the clinical application of 4DCT, motivating development of advanced 4DCT simulators.

Towards this goal, various approaches have been proposed to model lung inflation/deflation.
The first category of methods discretize the soft tissues (and bones) into masses (nodes) and connect them using springs and dampers (edges) based on mass-spring-damper system and CT scan values for spline-based MCAT phantoms (22), augmented reality based medical visualization (21), respiration animation (32), tumor motion modeling (27), and etc. A classical approach is to apply affine transformations to the control points to simulate respiratory motion. Lungs and body outline are linked to the surrounding ribs, such that they would have the synchronized expansion and contraction (22). These methods can only provide approximate lung deformations, and are intuitively simple and execute much faster than the accurate models based on continuum mechanics.

The second category of methods use hyperelastic models to describe the non-linear stress-strain behavior of the lung, such as, Ogden (16), Fung (29; 12), Yeoh (28), and polynomial (20). The straightforward way to simulate lung deformation between two breathing phases \( T_i, T_{i+1} \) is to use the lung shape at \( T_{i+1} \) as the contact/constraint surface and deform the lung at \( T_i \) based on the predefined mechanical properties of lung (24; 11). In this case, a negative pressure load on the lung surface is applied and Finite Element (FE) analysis is used to deform tissues (30). The lung will expand according to the negative pressure and slide against the contact surface to imitate the pleural fluid mechanism (4). This pressure can be estimated from the patient’s pleural pressure vs. lung volume curve, which in turn are measured from pulmonary compliance test (26). Along this line, patient-specific biomechanical parameters on the modeling process for FE analysis using 4DCT data are learned in (25). A deformable image registration of lungs study to find the optimum sliding characteristics and material compressibility using 4DCT data is presented in (1).

Besides lung deformation, the displacements of rib cage and diaphragm are also very important to design a realistic 4DCT simulator. Didier et al. (7) assume the rib cage motion is a rigid transformation and use finite helical axis method to simulate the kinematic behavior of the rib cage. They develop this method into a chest wall model (8) relating the ribs motion to thorax-outer surface motion for lung simulation. Saadé et al. (19) build a simple diaphragm model consisting of central tendon and peripheral muscular fibre. They apply cranio-caudal (CC) forces on each node of the muscular fibre to mimic the diaphragm contraction and use Cauchy-Green deformation tensor to describe the lung deformation. Hostettler et al. (13) consider internal organs inside the rib cage as a convex balloon and estimate internal deformation field directly through interpolation of the skin marker motions.

Patient-customized deformation approaches often assume a 4DCT of the patient is already available. We note that simulating deformations without any 4DCT has many challenges as lung motion changes considerably depending on health condition (with or without cancer), breathing pattern (abdomen vs. chest wall), age and many other factors. Nevertheless, thoracic deformation simulation without any prior (e.g. 4DCT of the same patient) is still very useful for developing treatment strategy in image-guided radiotherapy and generating controlled data to design and evaluate X-ray video based medical solutions.

In this paper, we present a biomechanical model based thoracic 4DCT simulation method that can faithfully simulate the deformation of lung and nearby organs for the whole breathing cycle. Our method takes only one CT scan as input, and defines the loads on the rib cage and the diaphragm to constrain the lung deformation. This differentiates our method from conventional continuum mechanics based algorithms. Our method can also simulate the passive mass-spring model based deformation of abdominal organs due to lung inflation/deflation. Conversion from density to mass assumptions for mass-spring model are supported by clinical data. To evaluate the accuracy of our simulator, we perform both qualitative image visual examination and quantitative comparison on expert annotated lung interior point pairs between multiple breathing phases, and demonstrate that our biomechanical model based simulation is very accurate.
Figure 1. Processing pipeline of our thoracic 4DCT simulator. There are two main stages in our simulation algorithm, lung deformation and 4DCT scan simulation. The tetrahedra on the cutting planes are colored in purple for demonstrating the quality of the volume mesh.

2. System Overview

Fig. 1 shows the processing pipeline of our 4DCT simulator based on biomechanical model. There are two main stages: Lung Deformation and 4DCT Scan Simulation. The first stage takes as input a thoracic CT scan and outputs the lung deformation at different respiratory phases, which are obtained through FE analysis of the lung biomechanical model. The second stage deforms the rest CT volume to generate the final 4DCT scans according to the previous lung deformation results. We model lung using a hyperelastic function based on nonlinear continuum mechanics such that it can give us accurate tumor motion trajectories inside the lung, and use mass-spring-damper model to deform the rest CT volume, thus, no segmentation of rig cage, spines, and other organs are necessary.

Our simulator generates 10~100 lung deformations between the maximum exhale phase $T_{ex}$ and the maximum inhale phase $T_{in}$, which in turn is used to synthesize the corresponding 4DCT phases. It can construct 4DCT phases with or without hysteresis of the tissue deformation, which basically defines how the lung volume is correlated with the inhale-to-exhale motion path and the exhale-to-inhale motion path. For no hysteresis, the CT’s for the same lung volume states are identical regardless of whether it is in the inhale-to-exhale cycle or the inhale-to-exhale cycle. For a given lung volume signal (as shown in the first row of Fig. 10), our simulator assembles the corresponding 4DCT phases to generate a sequence of CT’s. It can also embed synthetic tumors in the generated 4DCT as discussed in Sec. 5.

3. Biomechanical Simulation of Lung Deformation

3.1 Boundary Constraints Definition

For simplicity of notation, we use $x$, $y$, and $z$ to represent lateral, anterio-posterior (AP), and supero-inferior (SI) direction respectively. Since we do not assume we have a 4DCT of the patient...
available, it is not possible to use the actual lung surfaces of different breathing phases to define the deformation boundary constraints.

Instead, we define boundary constraints on the lung surface based on the anatomy and function of the human respiratory system (23) for the lung deformation. First, considering that the upper lobes of the lung are well constrained by the ribs, the displacement vectors (x, y and z components) of the tip surface region of upper lobes are fixed to avoid a pure translation of the lung when simulating the diaphragm contracting on the bottom. We take the clinical study in (10) as a basis for these constraints.

During inspiration, the lung sliding against the rib cage mainly occurs in the posterior/spine region, while in the anterior region, the lung expands with the increasing of thoracic cavity and the relative sliding between them is much smaller (5; 6). This phenomenon can also be observed in the DIR-Lab 4DCT dataset (2), which is one of the most recent clinical studies with expert annotations for this problem. Therefore, we define the boundary conditions for both the front and the back parts of the lung surface in order to simulate the different sliding actions. As shown in the boundary constraints box of Fig. 1, our system fixes the z displacement for all surface mesh vertices marked in red to simulate the coherent motion of lung with the thorax expansion on the axial plane. The selection of the vertices is based on empirical evidence (2). These vertices satisfy all these heuristics that they are on/near the convex hull of the lung surface, around the lateral sides of the middle and lower lobes, and have small (< 20°) normal variations.

To simulate the pleural sliding in the spine region, our simulator automatically locates the lung surface vertices in the vicinity of the thoracic vertebrae, and fixes the x and y displacements of these points as the third boundary constraint. Notice that our goal is to find surface vertices close to the spine, therefore we design a simple Gaussian curve fitting algorithm to locate these points instead of adopting a complicated thoracic vertebrae segmentation approach. The idea is to fit a set of Gaussian curves such that the area cut out by each curve is maximized. This provides a good global approximation to the spine shape and the constraint points can be accurately located. For simplicity, considering a sample 2D axial view, our algorithm maximizes the light blue region A covered by the blue Gaussian curve \( f(x) = ae^{-\frac{(x-b)^2}{2c^2}} \), as shown in Fig. 2(a).

We formulate it as a constrained multi-variable optimization problem as:

\[
\max_{a,b,c} \sum_{x=x_{\text{min}}}^{x_{\text{max}}} f(x), \quad \text{s.t. } f(x) - g(x) \leq 0, \forall x \in [x_{\text{min}}, x_{\text{max}}],
\]  

where the parameter \( a, b \) and \( c \) represent the scaling factor, expected value, and standard variance of \( f(x) \), \( x_{\text{min}} \) and \( x_{\text{max}} \) are the lung limits in the lateral direction, and \( g(x) \) is the upper limit for \( f(x) \) and is the minimum y value of the lung slice at each \( x \). In our simulator, this constrained
optimization problem is solved very efficiently by a sequential quadratic programming method, specifically active-set algorithm, which computes a quasi-Newton approximation to the Hessian of the Lagrangian at each iteration. We extend this 2D algorithm to the 3D CT volume by simply applying this algorithm slice by slice, as can be seen in Fig. 2 (b) and (c). Outliers occur in the top and bottom of the lung where \( g(x) \) is only partial constraints for the curve fitting. Our simulator removes these outliers by computing their difference to the mean Gaussian curve of the set, therefore correct fittings of the thoracic vertebrae are retained. The missing curves can be estimated by linear interpolation of the remaining curves. Note that to generate the binary CT slices for Gaussian curve fitting, we first convert the lung surface mesh to a solid volume, thus these CT slices can be obtained by intersecting the solid volume with a set of Axial planes.

### 3.2 Loads Definition

Since we are given one input CT scan and there is no bounding surface at the second breathing phase, we design an extra traction applied on the diaphragm area of the lung besides the negative intra-pleural surface pressure. The pressure force inflates the lung in all directions during inspiration, while the traction allows additional displacement in \( z \) direction to mimic the diaphragm contraction and pleural sliding.

Note that the pressure force can be well defined from the simulator input, therefore we focus on how to accurately locate the points (faces) that are close to the diaphragm for the definition of the traction. We model this as a graph search problem and solve it by our modified shortest closed-path algorithm. Our simulator first computes a dense 3D point cloud by finding the lung voxels at every \((x, y)\) location with the largest \( z \) value, as shown in Fig. 3(c), then converts the point cloud into a weight map (Fig. 3(d)) based on the local geometry information, and finally locates the diaphragm points (Fig. 4(e)) through our modified shortest closed-path algorithm. The left and right lower lobe are treated separately.

**Weight Map Definition:** We consider the 3D point cloud as an 2D image with intensity value from the \( z \) value of the corresponding point, and run the local Line Direction Discrepancy (LDD) computation on this image to generate the weight map \( W \). Thus our weight map computation can also be viewed as a special type of image filtering. As shown in Fig. 3(a), for each line \( d_i(x, y) \) of a \( 5 \times 5 \) block (the block size is set as \( 5 \times 5 \) for simplicity) centering at \((x, y)\), we build up two sub-lines \( d^1_i(x, y) \) and \( d^2_i(x, y) \) from 3-pixel tuples \((p^1_{i1}, p^1_{i2}, p^1_{i3})\) and \((p^2_{i1}, p^2_{i2}, p^2_{i3})\) respectively, \((i = 1, \ldots, 4)\), and compute the LDD as the minimum intersection angle of the four sub-line pairs. Alternatively, we compute the maximum of the cosine value of these angles to represent the weight, which can be efficiently calculated through dot product as

\[
W(p) = \max_{i=1,\ldots,4} \left\{ \frac{d^1_i \cdot d^2_i}{\|d^1_i\| \cdot \|d^2_i\|} \right\},
\]

where \( p \) represents pixel position \((x, y)\). Intuitively, regions with high curvature would have high/positive LDD value, for example, \( d_1 \) of \( B_3 \) in Fig. 3(b), while flat regions would have low/negative LDD values, for instance, \( B_1 \) and \( B_2 \).

**Diaphragm Point Segmentation:** Notice that all outliers locate at the boundary of the weight map, thus we formulate the diaphragm point segmentation as a shortest closed-path (SCP) problem, which finds a optimal cut along the boundary that separates the diaphragm points from the outliers.

To build the graph for SCP, we choose 4 neighborhood connection and set the edge weight \( E_{pq} \) as \( W(q) \). Therefore, \( E_{pq} \) and \( E_{qp} \) may have different weights. Instead of using the entire weight map to build the graph, we mask out the inner region through morphological operations and limit the optimal cut (red curve) between the inner \( \partial\Omega_2 \) and outer boundary \( \partial\Omega_1 \) (blur curves), as shown in Fig. 4(a). If we directly adopt the idea from (14) to design the SCP algorithm, some interior regions would be inevitably cut out to favor the lowest cost, as shown in Fig. 4 (a) and (b).

To solve this problem, we first sample the outer boundary \( \partial\Omega_1 \) every 10 points and find their
Figure 3. Weight map calculation for diaphragm point segmentation. (a) The line direction definition of our LDD measure, (b) sample blocks on the lung surface to illustrate our weight calculation algorithm, and (d) the weight map corresponding to the input point cloud (c). The orange region $d_1$ of $B_1$ has the highest LDD value out of the three sample blocks.

Figure 4. Diaphragm point segmentation. (a) The optimal cut by conventional SCP algorithm in red while the inner and outer boundaries are overlaid on the weight map in blue. (b) The estimated diaphragm points of (a) with respect to (c). Our new SCP algorithm unbends the ring regions in (c) into ribbon belts (f), and can accurately segment out the diaphragm points for traction definition in (d) and (e).

corresponding points (in terms of Euclidean distance) on the inter boundary $\partial \Omega_2$, as the green lines shown in Fig. 4(c). For the rest points on $\partial \Omega_1$, we compute their matches on $\partial \Omega_2$ (purple lines) through linearly interpolation of the previous matches (green lines), such that there are no crossing matches (lines) and correct ordering could be maintained. In this way, we can unbend the ring region between $\partial \Omega_1$ and $\partial \Omega_2$ into a ribbon belt (Fig. 4(f)) by aligning up all the purple and green lines in order, and set the length of the ribbon as the length of $\partial \Omega_1$ and the width as the shortest distance between $\partial \Omega_1$ and $\partial \Omega_2$. We then build up a new adjacency matrix/graph from the ribbon for the SCP algorithm. As we can see from Fig. 4(d-f), this would give us the accurate diaphragm points for the traction definition.

3.3 Finite Element Simulation

The final step for biomechanical simulation of lung deformation is to define the material property of the lung and apply FE analysis. We assume the lung tissue is homogeneous, isotropic, and use the first-order Ogden model (16; 17) to describe its non-linear strain energy density function as

$$W(\lambda_1, \lambda_2, \lambda_3, J) = \frac{\mu_1}{\alpha_1} (\lambda_1^{\alpha_1} + \lambda_2^{\alpha_1} + \lambda_3^{\alpha_1} - 3) + \frac{K}{2} (\ln J)^2,$$

where $\lambda_{1,2,3}$ are the deviatoric principal stretches, $\mu_1$ and $\alpha_1$ are material constants, $J$ is the Jacobian of the lung deformation, and $K$ is the bulk modulus chosen sufficiently high to satisfy near-incompressibility. Here, we assume that the shear modulus $\mu$ of the material is equal to $\mu_1 \alpha_1/2$, and we choose the Ogden parameters from (9) for all our experiments, that is, $\mu_1 = 0.0329$, and $\alpha_1 = 6.82$.

Next, we combine all the information (meshes, loads, and boundaries) defined in the previous sections into a single script file and directly run a FE solver to simulate the lung deformation. We integrate the open-source FEBio (9) into our simulator as the FE solver, and a lung deformation
Figure 5. Finite element analysis of a left lung deformation during inspiration. The top row displays the posterior view and the bottom row shows the inferior view. Color shows the degree of displacement with red denoting maximum displacement. One important reason that we choose FEBio is that it is specifically designed for biomechanical applications and offers constitutive models and boundary conditions that are relevant to the modeling of soft tissue deformation. The wide range of boundary interactions supported in FEBio makes the modeling of pleural sliding very straightforward.

4. 4DCT Scan Simulation

The deformation of the rest of the CT volume would be very complicated if we adopt a continuum mechanics based method, which requires segmenting out all the soft tissues, organs, bones, and other structures, and manually defining different material properties and their corresponding boundary constraints. Therefore, we treat the rest CT volume as a discrete structure of elements and model the rest CT volume deformation using the mass-spring-damper (MSD) system.

4.1 System Element Definitions

As shown in the bottom row of Fig. 1, we construct the MSD system by first sampling control points (masses) in the CT volume excluding the lung region. Then we connect these points through constrained Delaunay algorithm to form the links (springs), as the cutting plane demonstrated in the bottom center of Fig. 1. To define mass values for each control points, we first compute their physical densities from their CT image values, i.e., Hounsfield units (HU), and then the conversion from density to mass is very straightforward. The HU versus Electron/physical densities curve for human organs and tissues should be available after the specific CT device installation/calibration. We use the following approximate HU density conversion (3) for all our experiments,

\[
\rho(x) = \begin{cases} 
1.1e^{-3} \cdot h(x) + 1.060 & \text{if } h(x) < -13, \\
6.0e^{-4} \cdot h(x) + 1.081 & \text{otherwise},
\end{cases}
\]

where \( \rho(x) \) represents the physical density \( g/cm^3 \) at position \( x \), and \( h(x) \) HU value.

The guideline to define spring constants for the MSD system is to assign large values to bones, small values to soft tissues, lung, blood, air, and etc. based on their HU values. Considering the HU value for bones ranging from +700 (cancellable bone) to 3000 (dense bone), we nonlinearly map
the HU values [700, 3000] to spring constants [0.8, 1] based on the reciprocal of an exponential function, while the spring constants of other materials are mapped to [0.001, 0.8]. Other heuristic mappings are also possible yet have little impact onto the system performance.

### 4.2 Dynamics and Forces

We aim to deform the rest CT volume based on the lung deformation at each breathing phase, therefore, our system is a unforced version of a regular MSD system, that is, the external force $F_{ext} = 0$. The input is the prescribed displacements of the lung surface vertices, which generates the tensions for the internal springs by Hooke’s Law after each lung deformation. We also add damping force $F_d$ to reduce the amplitude of system oscillations. Thus from Newton’s 2\textsuperscript{nd} Law, the total force of the system $F_{tot}$ is equal to the internal force $F_{int}$, and defined as

$$F_{tot} = F_{int} = F_s + F_d = m \cdot \frac{d^2y}{dt^2},$$

where $F_s$ is the spring tension defined as $F_s = -k \cdot y$, $F_d = -R \cdot dy/dt$, $k$ is the $n \times 1$ spring constant vector, $y$ is the displacement vector of the $n$ control points ($n = 12000$ for our experiments), $m$ is the mass vector corresponding to the control points, and operator $\cdot$ denotes inner product of two vectors. We define the system as a critically damped system, therefore, the damping factor $R = 2\sqrt{m \cdot k}$. Thus Eq.5 can be derived as,

$$m \cdot \ddot{y} + R \cdot \dot{y} + k \cdot y = 0.$$

We solve this dynamic system using the finite difference method similar to (18). Between each iteration, we fix the motion of mass points covered by the spine region (Sec.3.1) to mimic the stationary spine movement. Once we have the deformation of the control points, we can compute the deformations of all voxels by trilinear interpolation. Other interpolation method, such as, tricubic interpolation, can also be employed to get smoother and continuous differentiable volumes.

### 5. Results and Discussion

**Volume Mesh Generation:** To generate the lung volume mesh for FE analysis of the lung deformation, we use TurtleSeg (http://www.turtleseg.org/) for the lung segmentation and TetGen (http://tetgen.berlios.de/) for the 3D tetrahedralization of the surface mesh. TurtleSeg can highlight the image regions where minimal user labeling could greatly improve the segmentation results. However, since it is for general purpose segmentation and not specialized for lung segmentations, additional mesh editing is required to fix small missing pieces and holes. In our experiments, the number of surface vertices is also reduced to $\sim 4000$ via quadric edge collapse decimation for simplification. We have incorporated TetGen into our simulator, and once the surface mesh is given, the rest of the lung simulation can run automatically.

One common problem of the surface mesh generated from segmentation tools is that the mesh is highly non-regular, and contains lots of thin triangles and over-sampled regions, as shown in Fig. 6(a). This would lead to badly shaped elements during the 3D Delaunay tetrahedralization process and hold back the fast convergence of biomechanical simulations. To optimize the obtained surface mesh, our simulator first converts it into a solid volume, and then computes a remeshed surface through marching cube algorithm (15) over the volume. This mesh optimization can also fix many other problems with the segmentation results, for instance, self-intersecting faces and non manifold edges/vertices. As we can see from Fig. 6(b), the optimized surface contains well-shaped triangles and uniformly sampled vertices, which constitute a valid basis for 3D Delaunay tetrahedralization.
Figure 6. Lung surface mesh optimization. a) shows a portion of the simplified surface mesh generated from TurtleSeg. b) shows our corresponding mesh optimization result. Most of the non-regular triangles and over-sampled regions are fixed.

Table 1. Mean error (and standard deviation) of the deformed lungs measured in 3D space and its x, y, and z components in mm. This table demonstrates that our biomechanical simulation algorithm for lung deformation is accurate and performs better than (13) on tested DIR-LAB 4DCT datasets (2).

<table>
<thead>
<tr>
<th>Case ID</th>
<th>CT Dims</th>
<th>Overall 3D x</th>
<th>Overall 3D y</th>
<th>Overall 3D z</th>
<th>Overall 3D Mean</th>
<th>Overall 3D Std</th>
<th>Our Results Mean</th>
<th>Our Results Std</th>
<th>Hostettler et al. (13) Mean</th>
<th>Hostettler et al. (13) Std</th>
</tr>
</thead>
<tbody>
<tr>
<td>case7</td>
<td>512 × 512 × 136</td>
<td>3.79 (1.80)</td>
<td>1.01 (0.96)</td>
<td>2.16 (1.57)</td>
<td>2.26 (1.85)</td>
<td>5.31 (3.35)</td>
<td>1.68 (1.03)</td>
<td>2.05 (1.32)</td>
<td>4.12 (3.54)</td>
<td></td>
</tr>
<tr>
<td>case8</td>
<td>512 × 512 × 128</td>
<td>6.15 (3.31)</td>
<td>2.01 (1.68)</td>
<td>2.58 (2.16)</td>
<td>4.24 (3.88)</td>
<td>10.81 (4.93)</td>
<td>2.06 (1.15)</td>
<td>5.31 (3.64)</td>
<td>7.39 (4.46)</td>
<td></td>
</tr>
<tr>
<td>case9</td>
<td>512 × 512 × 128</td>
<td>3.17 (1.37)</td>
<td>0.96 (0.80)</td>
<td>2.10 (1.33)</td>
<td>1.63 (1.23)</td>
<td>5.86 (1.83)</td>
<td>1.73 (1.65)</td>
<td>4.24 (1.44)</td>
<td>2.89 (2.29)</td>
<td></td>
</tr>
<tr>
<td>case10</td>
<td>512 × 512 × 120</td>
<td>1.37 (2.95)</td>
<td>1.08 (1.23)</td>
<td>1.24 (1.16)</td>
<td>3.59 (3.06)</td>
<td>6.93 (2.86)</td>
<td>3.05 (1.97)</td>
<td>4.92 (1.47)</td>
<td>6.16 (3.26)</td>
<td></td>
</tr>
</tbody>
</table>

Figure 7. Mean error distributions of our simulation results and Hostettler et al. (13) for overall 3D, and in x, y, and z directions for case7. Horizontal axes are the error magnitudes in mm. As visible, our simulator has more accurate estimation.

**Lung Deformation:** Fig. 5 shows an example of FE analysis of a left lung deformation during inspiration. The simulation results resemble the real 4DCT lung deformation with the maximum displacement occurring in the posterior region along the SI direction. The results also demonstrate realistic lung inflating effect due to the negative surface pressure, which can be better viewed in the second row of the figure. In our FE analysis, we define the simulation time for the inspiration phase is 2 seconds with step size $\Delta t = 0.1$, pressure force -0.02 and traction 0.005. For other parameters, for example, convergence tolerance, we use the default values in the FEBio solver.

To demonstrate the accuracy of our FE simulation, we also evaluate our simulator on the DIR-LAB dataset (2). We omit the down-sampled 4DCT cases of the DIR-LAB dataset and present our simulation results for case 7-10. Each test case has the same CT slice resolution of 512 × 512, and 300 manually labeled corresponding points between $T_{ex}$ and $T_{in}$. For case 7, the average landmark displacement is 11.59 ± 7.87 (standard deviation) mm, and the observer error is 0.81 ± 1.32 mm. Detailed specifications of the dataset can be found at http://www.dir-lab.com.

In our experiments, we compute the error as the Euclidean distance between our simulated displacement vectors and the manually labeled ones. We also implement the deformation filed...
estimation algorithm proposed by Hostettler et al. (13), and set its model parameters using the ground-truth marker displacement vectors. In this case, the simulation results of (13) are of the best performance that it can theoretically get, and we don’t have this ground-truth displacement information available for our simulator. We compare its simulation results with ours in Tab. 1, and the detailed distributions of simulation errors for case 7 in Fig. 7. From the table, we can see that the accuracy of our simulator improves roughly 40% compared with (13). The reasons why our simulator has larger errors in z direction are twofold. First, human lung generally has strong respiratory motions in this direction. And more importantly, the CT volume data has stronger artifacts and lower resolution in z than x and y, considering that the spatial resolution of tested CT data is $0.97 \times 0.97 \times 2.5$ mm.

In Fig. 8, we show the comparison between our FE analysis results and the ground-truth displacement vectors for case 7. For better illustration, we only show the left lung, which has 153 landmark points. We can see that our simulator works well in the lower posterior region where nodal displacement is mostly prominent. This indicates that our lung deformation simulator could provide valuable location-specific tumor motion information for physician to reduce the margins between clinical target volume (CTV) and planning target volume (PTV). We also notice that our simulation results have large angular difference with the manually labeled data in the upper anterior region. That is partially because of the lack of appropriate force definition for these elements in the simulator besides the negative surface pressure. It is also possible that the manually identified landmark points contain large errors since nodal displacement in this region is less than or around the z spatial resolution of the CT dataset.

In Fig. 9, we show the comparison between the performance of our simulation algorithm and Hostettler et al. (13). Both Fig. 9 and Tab. 1 demonstrate that our biomechanical simulation algorithm for lung deformation is very accurate considering the very low spatial resolution of the 4DCT dataset. We ran all these experiments on a linux (kernel 3.14.5) laptop with Intel i5-2520M CPU @ 2.50GHz and 6GB system memory using Matlab R2013b. As we mentioned before, for a typical 2 seconds simulation with step size $\Delta t = 0.1$ second, FEBio solver usually converges to the optimal solution in $\sim 31$ steps and takes about 115 seconds in our test environment. During the solving process, it in average runs through 278 equilibrium iterations for a volume mesh with 8650 nodes and 32793 solid elements. Please note that for each experiment, the simulation time varies since the convergence speed depends on the complexity of the input volume mesh, loads definitions, boundary conditions, and many other factors. For the detailed performance of FEBio solver, please refer to (9).

As indicated by (2), these test cases have very different patient lung shapes, tumor sizes and
Figure 9. Comparison between our simulation algorithm and Hostettler et al. (13) on case 8 (a), case 9 (b) and case 10 (c). For each test case (column), we choose two different viewpoints which can best show the difference between these two methods, and draw the displacement vectors at manually identified landmark positions. The blue lines represent (13)'s optimal displacement of the landmark points between $T_{ex}$ and $T_{in}$, while the red lines represent our simulation results.

locations, and breathing mechanisms. A simple interpolation between axial lung envelopes adopted by Hostettler et al. (13) inevitably generates large errors while our simulation algorithm adapts to different patients, thus achieves comparably more accurate results as shown in Tab. 1. We like to point out that our algorithm is a patient-customized lung deformation simulator. By providing more sophisticated constraints, its simulation quality will improve further. For instance, the patient lung surface in case-8 is heavily curved in the back/posterior region, thus including extra constraints to maintain this curved lung shape may make the simulation more precise. At last, the fidelity of the simulator may be improved by considering heart vibration models.

**4DCT Scan Simulation:** To generate simulated 4DCT scans, we treat the lung deformation as the equivalent force that drives the deformation of the CT volume since we have already considered all the influencing factors, for instance, the movement of body outline and diaphragm, into the biomechanical simulation of lung deformation. Therefore for every lung deformation, we can compute its corresponding CT volume deformation through our nonlinear mass-spring-damper system. That is, for a typical simulation of lung deformation as mentioned above, we already have 31 deformed lung volumes from FEBio solver, which would generate 31 deformed CT volumes using the technique presented in Sec.4.2. These 31 CT volumes represent the deformation of thoracic area during half breathing cycle (T00-T50).

For real-time 4DCT scan simulation, we first pre-process the half-cycle 31 CT volumes by transforming/interpolating them into N volumes ($N = 100$ in our simulation), and then approximate the real 4DCT scans by assembling the pre-computed N CT volumes based on the magnitude of the
Frames

Figure 10. Simulated tumor embedded 4DCT scans. The first row shows the normalized input breathing signal for our 4DCT simulation. The following 3 rows show our simulations at frame 154, 275, and 389 for CT slices on Axial, Coronal, and Sagittal planes, and X-ray images on Coronal and Sagittal planes. The ground truth tumor boundaries are overlaid on the X-ray images in blue. This 4DCT scan simulation is based on the lung deformation of Fig. 5.

In general, for each input lung deformation, our nonlinear mass-spring-damper system takes about \(33.76\) seconds to achieve the equilibrium state in \(\sim 500\) finite difference steps. Generating each of the \(N\) interpolated CT scans takes about \(166.5\) seconds, and once we have the pre-computed \(N\) CT scans, the 4DCT scan simulation runs at \(\sim 108\) fps (frames per second). Fig. 10 shows our 4DCT simulation results with respect to an input breathing signal of the patient\(^1\). From the figure, we can see that the abdominal organs are squeezed down during rib cage expansion, while the spines are fixed at the same time. This demonstrates that our simulator is capable of providing realistic deformation/displacement of liver, bones, and other soft tissues besides the accurate respiratory motion of lung.

**Tumor Embedding:** The right two columns of Fig. 10 show three pairs of X-ray images of coronal and sagittal views through integrating their corresponding CT volumes, and a sample tumor is embedded into the right lung and visible in the coronal view of the simulated CT scans. In our simulator, tumors can be easily embedded into the lung volume mesh through computing barycentric coordinates of their vertices to nearby lung tetrahedra. During X-ray images integration, we

\(^{1}\)Here we assume the normalized (to \([0, 1]\)) breathing signal is obtained from either spirometer or other types of measuring devices, such as magnetic sensors positioned on the epigastric region.
can synthesize the ground truth 3D tumor contours as well, as overlaid on the X-ray images in cyan. By putting different types of tumor at different positions of the lung, our simulator is able to provide unbiased ground truth for validating the accuracy of different tumor tracking algorithms (31). Compared with X-ray videos with metallic markers, our X-ray images remove the side effects of high contrast regions introduced by markers and can provide realistic images for tracking algorithms to extract faithful region features. Our simulator also eliminates the need for manually drawing ground truth tumor contours, which may contain large inter- and intra-physician discrepancies.

**Further Discussion:** Our simulator can be easily extended to simulate real-time patient-specific 4DCT scans, if given the synchronized data of pleural pressure, chest and abdomen height besides the single CT input. In this case, the FE analysis of the lung deformation would be converged very quickly since there is only a small difference of the lung shape between two time steps. The pleural pressure can be estimated from the lung volume readings through the pressure-volume curve, and it will give us the precise negative pressure for defining the deformation force. While the synchronized chest and abdomen height together give us the patient body outline during respiration, thus define the boundary constraints for both lung deformation and 4DCT scan simulation.

6. Conclusions

We have presented a biomechanical model based thoracic 4DCT simulation algorithm for examining the patient lung deformation induced by respiration given only one CT scan input. We model the lung stress-strain behaviour using a hyperelastic Ogden model, and treat the rest CT volume as discretized mass points connected by springs and dampers. This novel heterogeneous design leverages the advantages of both continuum mechanics and mass-spring-damper system in the way that the lung deformation is computed in very high accuracy while the deformation of the rest CT volume is achieved under practical computational constraints. Extensive analysis and comparisons with the manually labeled DIR-LAB dataset demonstrate that our lung deformation results are accurate.

References


