Research Article

3D-2D Deformable Image Registration Using Feature-Based Nonuniform Meshes

**Zichun Zhong,[1,2] Xiaohu Guo,[1] Yiqi Cai,[1] Yin Yang,[3] Jing Wang,[2]
Xun Jia,[2] and Weihua Mao[2]**

[1]*University of Texas at Dallas, Richardson, TX 75080, USA*
[2]*University of Texas Southwestern Medical Center, Dallas, TX 75235, USA*
[3]*University of New Mexico, Albuquerque, NM 87131, USA*

Correspondence should be addressed to Weihua Mao; weihuamao@gmail.com

By using prior information of planning CT images and feature-based nonuniform meshes, this paper demonstrates that volumetric images can be efficiently registered with a very small portion of 2D projection images of a Cone-Beam Computed Tomography (CBCT) scan. After a density field is computed based on the extracted feature edges from planning CT images, nonuniform tetrahedral meshes will be automatically generated to better characterize the image features according to the density field; that is, finer meshes are generated for features. The displacement vector fields (DVFs) are specified at the mesh vertices to drive the deformation of original CT images. Digitally reconstructed radiographs (DRRs) of the deformed anatomy are generated and compared with corresponding 2D projections. DVFs are optimized to minimize the objective function including differences between DRRs and projections and the regularity. To further accelerate the above 3D-2D registration, a procedure to obtain good initial deformations by deforming the volume surface to match 2D body boundary on projections has been developed. This complete method is evaluated quantitatively by using several digital phantoms and data from head and neck cancer patients. The feature-based nonuniform meshing method leads to better results than either uniform orthogonal grid or uniform tetrahedral meshes.

1. Introduction

Cone-Beam Computed Tomography (CBCT) has been widely used for accurate patient setup (initial positioning) and adaptive radiation therapy. Attentions are still needed to reduce imaging radiation doses and improve image qualities. Traditionally, Conventional Cone-Beam Computed Tomography (CBCT) image reconstruction in radiation therapy needs hundreds of projections, which deliver high imaging dose to patients. In order to reduce the number of CBCT projections, recently, some researchers have proposed methods to reconstruct images by using the information of prior images, such as a planning CT [1–3] or a previous CBCT [4], and a deformation model, which is essentially a 3D-2D deformable image registration (DIR) procedure. A lot of researches have been done related to 3D-2D image registration. Previously, only rigid registrations between 3D images

and 2D fluoroscopic images have been addressed [5–7]. For 3D-2D nonrigid registration [1, 2, 4, 8, 9], the multiscale technique was applied, instead of using the finite element method- (FEM-) based methods to speed up the reconstruction process and increase the accuracy. For nonrigid modeling of respiratory motion, Zeng et al. [10] introduced a method to estimate the 3D motion parameters of a nonrigid, free breathing motion model from a set of projection views. In order to improve the computational efficiency, Jia et al. [11] developed a GPU-based algorithm to reconstruct high-quality CBCT images from undersampled and noisy projection data so as to lower the imaging dose, but it does not make use of the planning CT. However, in all these methods, voxel-based deformation fields were employed to estimate a large number of unknowns, which required extremely long computational time. Additionally, in their deformation models, the image features and organ boundaries were not

specifically considered, which may cause inaccurate deformation estimation. In this work, we proposed a new FEM-based approach, that is, a feature-based nonuniform meshing method, to overcome these limitations.

FEM can be best understood from its practical application, for instance, mesh discretization of a continuous domain into a set of discrete subdomains. It has already been used in image registration [12, 13]. Usually, FEM is exploited to achieve two important advantageous aspects. On one hand, it endows the efficiency of the registration process due to a small number of sampling points compared with voxel-based sampling methods. On the other hand, it provides the smoothness of the displacement vector field (DVF) due to the smoothness constraint between elements and the interpolation within one element. The quality of geometric discretization is crucial for the effectiveness of the image registration applications. Surface meshing methods in 3D-3D image registration [13–15] and volume meshing methods in 3D-3D image registration [16–19] and in 3D image reconstruction [20, 21] have been applied, but none of them were employed in 3D-2D DIR.

When 3D-2D DIR algorithm is used iteratively to reconstruct 3D volumetric images, the number of sampling points is crucial for the computation. A large number of sampling points could lead to a very slow computational speed, while a limited number of points with uniform distribution could miss some important image features and make the registration less accurate. In our proposed method, a special FEM system is developed to automatically generate high-quality adaptive meshes conforming to the image features for the whole volume without user's manual segmentation. This system allows for more sampling points placed in important regions (at organ/tissue/body boundaries or regions with highly nonlinearly varying image intensity); while fewer sampling points are placed within homogeneous or in regions with linearly varying intensity. In this way, deformations of boundaries and other important features can be directly characterized by the displacements of the sampling points that are lying on boundaries or features, rather than interpolating from a uniform grid or a larger-sized tetrahedron in the volume mesh. As a result, the deformation can be controlled more precisely. With approximately the same numbers of sampling points, the feature-based nonuniform meshing method produces better deformed volumetric images comparing with methods using uniform meshes. The high-quality digitally reconstructed radiographs (DRRs) of the deformed anatomy are generated by using ray tracing method. Subsequently, these DRRs are compared with corresponding 2D projections from CBCT scans, and the DVF is optimized iteratively to obtain the final reconstructed volumetric images.

In order to provide a good initial DVF and accelerate the calculation, we proposed a boundary-based 3D-2D DIR method before the aforementioned 3D-2D DIR. Although researches on boundary-guided (or contour-guided) image registrations [22–24] have been carried out for many years, their methods were applied on either 2D-2D or 3D-3D DIR cases. Our proposed algorithm is suitable to employ on 3D-2D DIR, while dealing with large deformations for adaptive radiation therapy.

This paper makes the following contributions for effectively computing 3D-2D DIR:

(i) Compared with the traditional voxel-based method, the mesh-based methods have faster computational speed and better DVFs, since the voxel-based deformations were employed to estimate a large number of unknowns, which required extremely long computational time and were easy to be trapped in localized deformations.

(ii) When equal numbers of sampling points are used, the nonuniform meshing method leads to obtaining higher quality reconstructed images and better DVF compared with that of uniform meshes under the same number of optimization iterations.

(iii) Due to the large data sizes of the volume and projection images, the boundary-based DIR technique and GPU-based parallel implementation have been applied and achieved high computational efficiency and the reconstruction of $512 \times 512 \times 140$ CBCT image can be done within 3 minutes, which is close to clinical applications.

The rest of the paper is organized as follows. Section 2 describes the proposed methods and materials in detail, including nonuniform mesh generation, the framework of the nonuniform meshing to reconstruct volumetric images by using 3D-2D DIR, and the boundary-based 3D-2D DIR. In Section 3, the experimental results are discussed to evaluate the proposed methods qualitatively and quantitatively. Several digital phantoms and patient data sets are measured. Finally, the conclusion and future work are given in Section 4.

2. Methods and Materials

Figure 1 illustrates a flow chart of the entire proposed technique. The dash-line box is used to provide initial DVFs and accelerate the calculation by proposed boundary-based 3D-2D DIR method, which will be described in Section 2.3. All other parts illustrate how the proposed novel nonuniform-mesh-guided 3D-2D image registration method is used to deform the original planning CT images. This requires much smaller degrees of freedom to generate the DVF than that of voxel-based methods. Each step is described in the following subsections and we will firstly introduce the nonuniform-mesh-guided 3D-2D image registration method and then discuss the boundary-based 3D-2D DIR method.

2.1. Creation of Nonuniform Meshes. After the user specifies the total number of mesh vertices, a feature-based nonuniform tetrahedral mesh is generated automatically. Nonuniform meshes are important for improving the accuracy of the numerical simulations as well as better approximating the shapes. Zhong et al. [25] developed a novel particle-based nonuniform surface meshing approach by formulating the interparticle energy optimization in a fast convergence technique. In this method, users will design a density field, which is used to control the distribution of the particles (sampling points). This particle-based surface meshing framework is
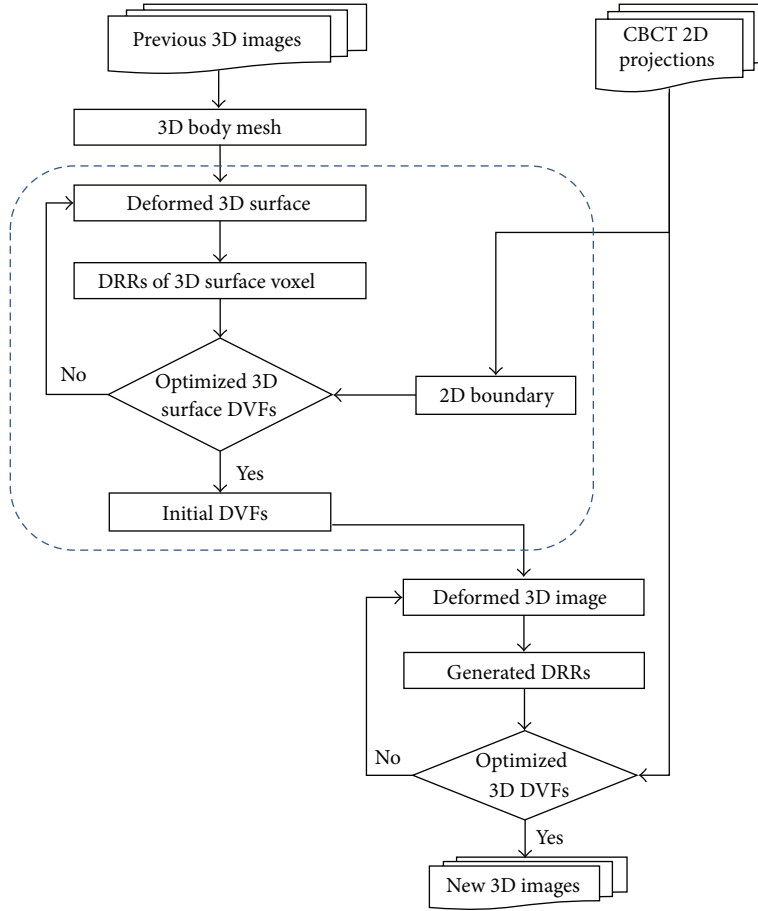
FIGURE 1: Flow chart of the proposed 3D-2D image registration method. The dash-line box is used to provide initial DVFs and accelerate the calculation by proposed boundary-based 3D-2D DIR method. It may be skipped according to different scenarios. More details are given in Section 2.3.

extended to 3D volume case in this paper so as to generate tetrahedral meshes based on 3D volume images.

2.1.1. Basic Meshing Framework. The basic idea of the mesh generation is similar to Zhong et al.'s work [25], and the main difference is that we extend their work to 3D nonuniform volume meshing. There are two steps in mesh generation: particle optimization and Delaunay triangulation computation.

Regarding each mesh vertex as a particle, the potential energy between the particles decides the interparticle forces. When the forces on each particle reach equilibrium, particles arrive at an optimal balanced state, resulting in a uniform distribution. In this case, an isotropic meshing can be generated. To handle the adaptive meshing, the concept of "embedding space" [26, 27] is applied. In the Nash embedding theorem, it is stated that every Riemannian manifold [28] can be isometrically embedded into some high-dimensional Euclidean space. In such high-dimensional embedding space, the metric is uniform and isotropic. When the forces applied on each particle become equilibrial in this embedding space, the particle distribution in the original domain will exhibit the desired adaptive property, that is, conforming to

the user-defined density field. This property is used to formulate the particle-based adaptive meshing framework. The following concepts of density field, interparticle energy, and force are defined based on [25].

The density field is defined by using the following metric tensor as

$$\mathbf{M}(\mathbf{v}) = \rho(\mathbf{v})^{2/m} \cdot \mathbf{I}, \qquad (1)$$

where $\mathbf{v}$ is the particle position. $\mathbf{I}$ is the 3×3 identity matrix. $\mathbf{M}(\mathbf{v})$ defines an isotropic adaptive metric with the user-defined density function $\rho(\mathbf{v})$. m is the dimension of the original volume space, so $m = 3$.

Given N_v particles with their positions $\mathbf{V} = \{\mathbf{v}_i \mid i = 1 \cdots N_v\}$ in the volume Ω ($\Omega \in \mathbb{R}^m$) which is embedded in $\mathbf{R}^{\overline{m}}$ space, denoted as $\overline{\mathbf{V}} = \{\overline{\mathbf{v}}_i \mid i = 1 \cdots N_v\}$, where $m \leq \overline{m}$, the interparticle *energy* between particles i and j in such embedding space is defined as

$$\overline{E}^{ij} = e^{-\|\overline{\mathbf{v}}_i - \overline{\mathbf{v}}_j\|^2/4\sigma^2} = e^{-(\mathbf{v}_i - \mathbf{v}_j)^T \mathbf{M}_{ij}(\mathbf{v}_i - \mathbf{v}_j)/4\sigma^2}, \qquad (2)$$

where $\mathbf{M}_{ij}$ is the metric tensor between particles i and j, and, for simplicity, it is approximated by the average of metric tensors at two positions: $\mathbf{M}_{ij} \approx (\mathbf{M}(\mathbf{v}_i) + \mathbf{M}(\mathbf{v}_j))/2$.

Data: a 3D volume Ω with density field $\rho(\mathbf{v})$, and the desired number of vertices N_v
Result: an adaptive sampling $\mathbf{V}$ of Ω
Initialize particle locations $\mathbf{V}$;
While *stopping condition not satisfied* **do**
    Update the k-d tree [22] data structure for the current sampling locations $\mathbf{V}$;
    for *each particle i* **do**
        Obtain particle i's k-nearest neighbors $N(i)$ within five standard deviations (5σ) from k-d tree;
        for each particle $j \in N(i)$ **do**
            Compute $\overline{E}^{ij}$ by (2);
            Compute $\overline{\mathbf{F}}^{ij}$ by (4);
        end
        Compute the total force $\overline{\mathbf{F}}^i$ by (5);
    end
    Compute the total energy $\overline{E}$ by (3);
    Run L-BFGS algorithm with $\overline{E}$ and $\{\overline{\mathbf{F}}^i\}$, to update the particle locations $\mathbf{V}$;
    Project $\mathbf{V}$ onto the 3D volume surface, if $\mathbf{V}$ jumps out of the volume boundary;
end

ALGORITHM 1: Particle optimization with density field $\rho(\mathbf{v})$.

The exponent in the term $\overline{E}^{ij}$ can be approximated by $\|\overline{\mathbf{v}}_i - \overline{\mathbf{v}}_j\|^2 \approx (\mathbf{v}_i - \mathbf{v}_j)^T \mathbf{M}_{ij}(\mathbf{v}_i - \mathbf{v}_j)$. The interparticle energy as defined in (2) depends on how to choose the fixed kernel width σ. If σ is chosen too small, then particles will nearly stop spreading because there are almost no forces between particles. If σ is chosen too large, then nearby particles cannot repel each other and the resulting sampling distribution will be poor. From our extensive experiments, we find out that the best adaptive mesh quality can be achieved when Gaussian kernel width is set as $\sigma = 0.3\sqrt{|\overline{\Omega}|/N_v}$. $|\overline{\Omega}|$ is the image volume in the embedding space.

The total energy can be computed by summing up all pairs of interparticle energies:

$$\overline{E} = \sum_{i=1}^{N_v} \sum_{j=1, j \neq i}^{N_v} \overline{E}^{ij}. \tag{3}$$

The gradient of $\overline{E}^{ij}$ can be considered as the *force* $\overline{\mathbf{F}}^{ij}$ in the embedding space:

$$\overline{\mathbf{F}}^{ij} = \frac{\partial \overline{E}^{ij}}{\partial \overline{\mathbf{v}}_j} = \frac{(\overline{\mathbf{v}}_i - \overline{\mathbf{v}}_j)}{2\sigma^2} e^{-\|\overline{\mathbf{v}}_i - \overline{\mathbf{v}}_j\|^2/4\sigma^2}$$

$$= \frac{\mathbf{Q}_{ij}(\mathbf{v}_i - \mathbf{v}_j)}{2\sigma^2} e^{-(\mathbf{v}_i - \mathbf{v}_j)^T \mathbf{M}_{ij}(\mathbf{v}_i - \mathbf{v}_j)/4\sigma^2}, \tag{4}$$

where $\mathbf{Q}_{ij} = \sqrt{\mathbf{M}_{ij}}$. The details of the mathematical derivations are given in Sec. 3.2.2 of [25].

Then the total force applied on particle i is

$$\overline{\mathbf{F}}^i = \sum_{j \neq i} \overline{\mathbf{F}}^{ji}. \tag{5}$$

In the particle optimization algorithm, user can specify a density field $\rho(\mathbf{v})$ and desired number of vertices N_v. In

our implementation, for each particle, we only compute the mutual effects (i.e., energy and forces) from the particles within a neighborhood of five standard deviations (5σ); otherwise, the particles have no mutual effects due to the large distance between each other. The k-d tree [29] is a space-partitioning data structure for organizing points in a k-dimensional space and can quickly search such neighborhoods. With the total interparticle energy (3) and force (5), L-BFGS [30] (a quasi-Newton algorithm) optimization method is used to obtain the optimized adaptive particle positions. This optimization proceeds iteratively until convergence by satisfying a specified stopping condition; for example, the magnitude of the gradient or the maximal displacement of particles is smaller than a threshold, or the total number of iterations. Algorithm 1 shows the details of the adaptive particle optimization on 3D volume.

After optimizing the particle positions, the final desired nonuniform tetrahedral mesh can be generated by using the Delaunay triangulation [29]. If the density field is uniform in the entire volume, we can generate the isotropic tetrahedron mesh, which is used in Section 3 for comparison experiments.

2.1.2. Feature-Based Nonuniform Meshing on Medical Image. Figure 2 illustrates the feature-based nonuniform mesh generation on a set of torso images acquired from a digital phantom XCAT [31]. The 4D XCAT provides an accurate representation of complex human anatomy and has the advantage that its organ shapes can be changed to realistically model anatomical variations and patient motions; more importantly, it also provides voxel-based DVFs, which are used as the ground truths to evaluate the accuracy of the deformation.

It is necessary to design a density field to match the volume image features. Original images are preanalyzed using a Laplacian operator (searching for zero crossings in the second
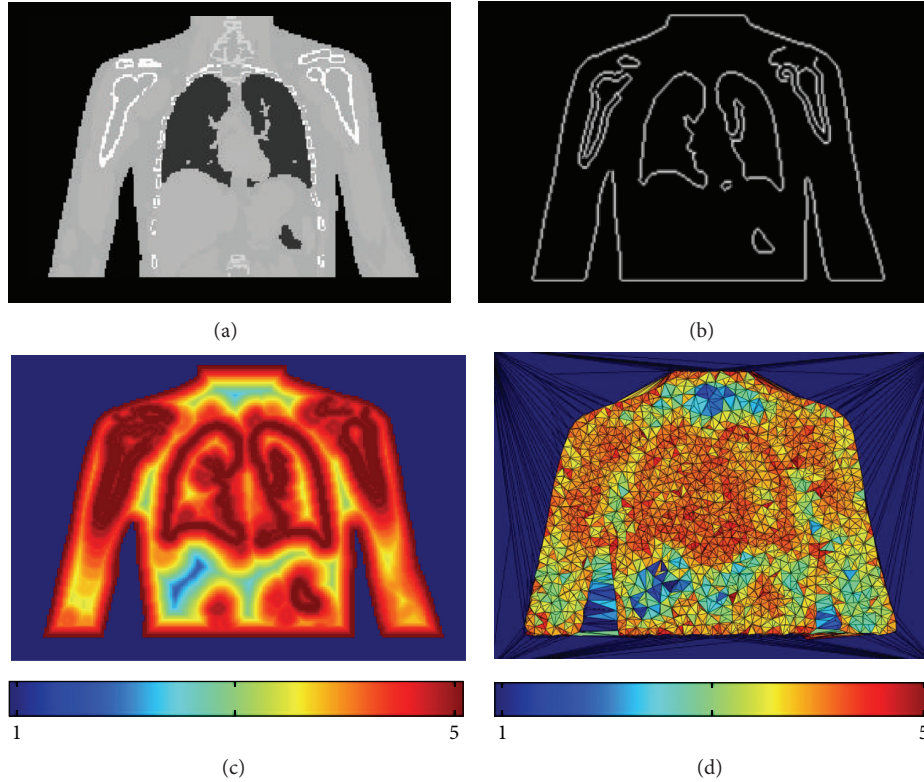
FIGURE 2: Demonstration of the feature-based nonuniform mesh generation on a digital XCAT phantom. (a) The original image; (b) extracted feature edges; (c) density field; (d) a 2D view of the interior meshes with color-mapping.

derivative of the image to find edges) to extract features including contour edges and boundaries between organs and tissues, which are regions with highly nonlinearly varying image intensities. Since the Laplacian operator as high-pass operator highlights edges as well as noise, it is desirable to smooth the image in order to suppress noises at first. When the feature edges of the volume image are obtained as shown in Figure 2(b), a density field could be calculated automatically without manual segmentation (Figure 2(c)). The density function $\rho(\mathbf{v})^{1/3}$ depends on the distance between the feature edges and the voxels in the volume domain. In our experiments, the smooth density field is defined as a piecewise linear function:

$$\rho(\mathbf{v})^{1/3} = \begin{cases} 5, & 0 \leq \varphi \leq 2, \\ \dfrac{49}{9} - \dfrac{2}{9}\varphi, & 2 < \varphi \leq 20, \\ 1, & \text{otherwise,} \end{cases} \tag{6}$$

where φ is the distance between the feature edges and the voxels measured by the voxel grid unit. The red color means the higher density field area, while the blue color means the lower density field area. The user can choose any other density functions according to their requirement. The motivation of designing the density field as a piecewise linear function is to make the density field as smooth as possible, so that the volume sizes of the tetrahedrons in the computed mesh can be controlled to vary smoothly. Finally, we can generate adaptive

meshes with high-quality tetrahedral elements. Equation (6) is given under our extensive DIR experiments, which is based on the volume image resolutions and voxel scales: if the sampling point is located within 2-voxel distance with respect to its nearest feature edge point, the density value is 5; if the sampling point is located between 2-voxel and 20-voxel distance with respect to its nearest feature edge point, the density value is computed based on the designed linear function in (6); if the sampling point is located beyond 20-voxel distance with respect to its nearest feature edge point, the density value is 1. From Figure 2, we can see that the designed density field can generate good tetrahedral meshes, which can well conform to the image features as well as obtain high-quality tetrahedral meshes.

After designing the density field, a binary mask needs to be computed from the original image by setting "one" inside of the human anatomy and "zero" outside to constrain the vertex positions inside or on the body during mesh vertices optimization. The mesh vertices are automatically computed by Algorithm 1. Vertices are densely positioned in the regions with highly nonlinearly varying image intensities, while regions of constant or linearly varying image intensities are assigned fewer vertices. Following this process, vertex locations are optimized conforming to the density field as illustrated in Figure 2(c). In order to control the entire volume image more effectively, 8 bounding box vertices of the human anatomy are added (as shown in Figure 3). Then volume meshes (tetrahedrons) are created based on the Delaunay
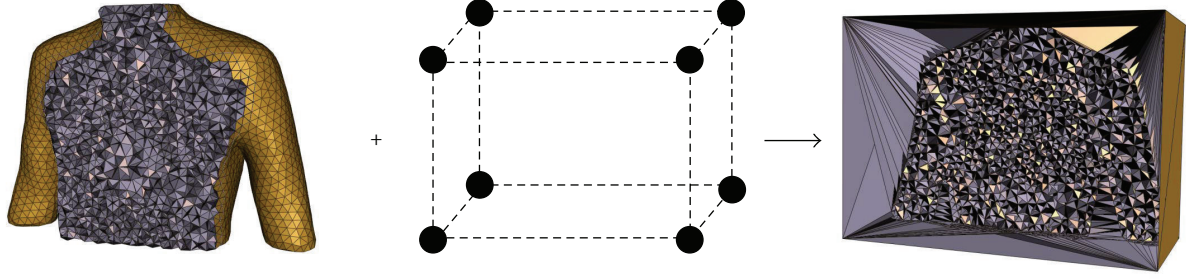
FIGURE 3: In order to control the entire volume image more effectively, 8 bounding box vertices of the human anatomy are added.

triangulation of the vertices. As a result, meshes corresponding to boundaries between organs and tissues are denser. The color-coded tetrahedrons of the generated feature-based mesh in Figure 2(d) illustrate that the tetrahedral volumes are well conforming to the desired density defined by the features of the given image. Due to the importance of deformations occurring around boundaries between organs and tissues, it is necessary to place more vertices (or sampling points) at features, while placing fewer vertices in nonfeature regions. In this way, if DVF is specified to each mesh vertex (or sampling point), the boundaries and other important features can be directly represented by the displacements of sampling points or represented by smaller tetrahedrons, rather than interpolating through four vertices of one larger-sized tetrahedron; then the deformation can be diffused from the mesh vertices to each voxel of the volume more accurately. This is the most significant advantage of the feature-based nonuniform meshing method.

2.2. Volumetric Image Reconstruction by 3D-2D DIR.
This section introduces how to use our generated feature-based nonuniform meshing to reconstruct high-quality volumetric images by using 3D-2D DIR.

2.2.1. Computation of Deformed Volume.
The displacement vector of each voxel ($\mathbf{D}^v$) is obtained through interpolating the DVF on mesh vertices ($\mathbf{D}$) by using the barycentric coordinates [32] of each voxel in its corresponding tetrahedron, and the deformed volume is resampled onto a uniform grid volume. The intensity of each voxel in the new deformed image U_{new} is calculated from the original CT image U_{original} according to $\mathbf{D}^v$ as follows:

$$U_{\text{new}}(a,b,c) = U_{\text{original}}\left(a + D_1^v(a,b,c), b + D_2^v(a,b,c), c + D_3^v(a,b,c)\right), \tag{7}$$

where D_1^v, D_2^v, and D_3^v are three floating point spatial components of the displacement vector ($\mathbf{D}^v$) along x, y, and z directions. The displacement vector points from the center of voxel (a,b,c) in the new deformed image to a point at $(a + D_1^v(a,b,c), b + D_2^v(a,b,c), c + D_3^v(a,b,c))$ in the original source image that will unlikely be at a voxel center. Then the original source intensity at that point is obtained via trilinear interpolation from its eight neighboring voxels.

This technique not only gives an accurate CT intensity to map to the new deformed image, but also acts as an antialiasing technique to avoid artifacts in the projected DRRs, since it is easy and efficient to compute DRRs from a uniform grid sampled volume by using ray tracing algorithm.

2.2.2. Computation of DRR Using Ray Tracing Algorithm.
Ray tracing is a technique for generating an image by tracing the path of light through pixels in an image plane and simulating the effects of its encounters with objects. In the DRR generation, the Siddon ray tracing algorithm is applied [33].

To better simulate the realistic raw target CBCT projections from XCAT phantom data and test the sensitivity of our method to the realistic complications, after the noise-free ray line integrals p_i are computed, the noisy signal I_i at each pixel i is generated based on the following noise model:

$$I_i = \text{Poisson}\left(I_0 e^{-p_i}\right) + \text{Normal}\left(0, \sigma_e^2\right), \tag{8}$$

where Poisson is Poisson distribution and Normal is normal distribution. I_0 is the incident X-ray intensity and σ_e^2 is the background electronic noise variance. In this study, I_0 is set to 1×10^5 and σ_e^2 is set to 10 [34, 35].

2.2.3. Optimization of 3D-2D DIR Energy.
The deformation is optimized by minimizing the total energy E, which includes two terms, the regularization ($E_{\text{reg}}(\mathbf{D})$) used to achieve smoothness of the DVF and the similarity ($E_{\text{sim}}(\mathbf{D})$) between the two images:

$$E(\mathbf{D}) = E_{\text{reg}}(\mathbf{D}) + E_{\text{sim}}(\mathbf{D})$$
$$= \mu L(\mathbf{D}) + \sum_{m=1}^{N_p} \left\| R(\mathbf{D}, \theta_m) - I(\theta_m) \right\|^2, \tag{9}$$

where μ is a weighting factor to control the tradeoff between the similarity and regularization. It is empirically set at 10.0 for all of the experiments in this paper. $\mathbf{D}$ is the DVF defined on the mesh vertices. $L(\mathbf{D})$ is regularization term defined as

$$L(\mathbf{D}) = \sum_{i=1}^{N_v} \sum_{d=1}^{3} \left[\left(\frac{\sum_{j\in N(i)} \left(\mathbf{D}_d(j) - \mathbf{D}_d(i) \right)}{|N(i)|} \right)^2 \right], \tag{10}$$

where $L(\mathbf{D})$ is a summation of the square of Graph Laplacian operations [36] on the DVF over every vertex except those on the external borders. $\mathbf{D}_d$ ($d = 1, 2, 3$) are three components of DVFs. N_v is the total number of the mesh vertices. $N(i)$ is the set of one-ring neighboring vertices (j) of vertex i. $|N(i)|$ is the size of set $N(i)$.

The second term of the energy function in (9) represents the similarity between the CBCT projections ($I(\theta_m)$) acquired at gantry angle (θ_m) beforehand and the DRRs ($R(\mathbf{D}, \theta_m)$) created after the deformation of $\mathbf{D}$ applied on planning CT at the same gantry angle. N_p is the number of projections involved. This term is a summation of the square of intensity difference over every pixel of all projections. Although R and I are two-dimensional in reality, we can easily use one-dimensional arrays to represent them for computational simplification.

L-BFGS algorithm [30] is used to optimize the DVF ($\mathbf{D}$). The gradient of the energy function E with respect to $\mathbf{D}$ can be calculated as follows:

$$\nabla E(\mathbf{D}) = \mu \nabla L(\mathbf{D})$$
$$+ 2 \sum_{m=1}^{N_p} \left(R(\mathbf{D}, \theta_m) - I(\theta_m) \right) \nabla R(\mathbf{D}, \theta_m). \tag{11}$$

For each iteration of L-BFGS optimization, the energy E and its gradient ∇E are updated.

The DRRs ($R(\mathbf{D}, \theta_m)$) are generated from the resampled deformed planning CT (U_{new}) by the ray tracing method for each gantry angle as

$$R(\mathbf{D}, \theta_m) = \mathbf{P}(\theta_m) U_{\text{new}}(\mathbf{D}), \tag{12}$$

where $\mathbf{P}(\theta_m)$ is the cone-beam projection matrix that describes the X-ray projection operations. The element $\omega^{\alpha,\beta}$ of matrix $\mathbf{P}$ is the weight of voxel β in U_{new} contributed to the pixel α in DRR during the projection simulation calculated by the ray tracing method.

Using (12), the gradient of the energy function E of (11) becomes

$$\nabla E(\mathbf{D}) = \mu \nabla L(\mathbf{D}) + 2 \sum_{m=1}^{N_p} \mathbf{P}(\theta_m)$$
$$\cdot \left(\mathbf{P}(\theta_m) U_{\text{new}}(\mathbf{D}) - I(\theta_m) \right) \nabla U_{\text{new}}(\mathbf{D}). \tag{13}$$

2.2.4. GPU-Based Acceleration.

The entire process of this volumetric image reconstruction method was implemented on GPU. The GPU card used in our experiments is an NVIDIA GeForce GTX 780 Ti with 3 GB GDDR5 video memory. It has 2,880 CUDA cores with a clock speed of 1,006 MHz. Utilizing such a GPU card with tremendous parallel computing ability can significantly increase the computation efficiency. There are two time-consuming processes during the reconstruction. One is the DRR generation, and the other is the gradient computation of the similarity term in the total energy E.

(1) DRR Generation on GPU. For the DRR generation part, it is straightforward to accomplish the ray tracing algorithm in parallel computation. For example, each pixel intensity of the DRR is determined by accumulating all of the weighted voxel intensities through which one X-ray goes. This computation process is highly independent between each ray line. In this case, different GPU threads can compute each ray line simultaneously without conflict.

(2) Computation of Energy Gradient on GPU. From (13), it can be seen that there are two terms in the gradient of the energy. One is the gradient of regularization $\mu \nabla L(\mathbf{D})$ with respect to DVF $\mathbf{D}$: this can be easily computed in parallel based on each mesh vertex. The other is the gradient of the similarity term. In order to demonstrate clearly how to compute this term in parallel, the second part of (13) is rewritten in more detail based on pixel α in DRR and voxel β in U_{new} at one projection in angle (θ_m). The gradient of the similarity term with respect to displacement vector of vertex i (i.e., $\mathbf{D}_i$) can be denoted as follows:

$$\frac{\partial E_{\text{sim}}(\mathbf{D}, \theta_m)}{\partial \mathbf{D}_i}$$
$$= 2 \sum_{\beta=1}^{|U_{\text{new}}(\mathbf{D},i)|} \sum_{\alpha=1}^{|R(\beta)|} \omega^{\alpha,\beta} \left(R^\alpha - I^\alpha \right) \nabla U_{\text{new}}^\beta(\mathbf{D}, i), \tag{14}$$

where $U_{\text{new}}(\mathbf{D}, i)$ is the set of voxels controlled by vertex i. $R(\beta)$ is the set of pixels on DRR affected by voxel β. $|U_{\text{new}}(\mathbf{D}, i)|$ and $|R(\beta)|$ are the sizes of the corresponding sets. R^α is the intensity of pixel α on DRR. I^α is the intensity of pixel α on CBCT projection. $\omega^{\alpha,\beta}$ is one element in the projection matrix $\mathbf{P}(\theta_m)$, that is, the weight of voxel $\beta(U_{\text{new}}^\beta(\mathbf{D}, i))$ contributed to the pixel α in DRR generation. $\nabla U_{\text{new}}^\beta(\mathbf{D}, i)$ is the gradient of voxel $\beta(U_{\text{new}}^\beta(\mathbf{D}, i))$ with respect to $\mathbf{D}_i$.

Then the gradient of the similarity term, that is, (14), can be rewritten by simplification as

$$\frac{\partial E_{\text{sim}}(\mathbf{D}, \theta_m)}{\partial \mathbf{D}_i} = \sum_{\beta=1}^{|U_{\text{new}}(\mathbf{D},i)|} \frac{\partial E_{\text{sim}}(\mathbf{D}, \theta_m)}{\partial U_{\text{new}}^\beta(\mathbf{D}, i)} \frac{\partial U_{\text{new}}^\beta(\mathbf{D}, i)}{\partial \mathbf{D}_i}. \tag{15}$$

Finally the total gradient for all projections with respect to displacement vector $\mathbf{D}_i$ is

$$\frac{\partial E_{\text{sim}}(\mathbf{D})}{\partial \mathbf{D}_i} = \sum_{m=1}^{N_p} \sum_{\beta=1}^{|U_{\text{new}}(\mathbf{D},i)|} \frac{\partial E_{\text{sim}}(\mathbf{D}, \theta_m, \beta)}{\partial \mathbf{D}_i}, \tag{16}$$

where N_p is the number of projections.

Now it is clearly shown that there are two components in the gradient of the similarity term in (15).

(a) Gradient computation of the similarity energy with respect to voxel intensity ($\partial E_{\text{sim}}(\mathbf{D}, \theta_m)/\partial U_{\text{new}}^\beta(\mathbf{D}, i)$): this can be computed when the DRRs are generated and then stored in a volume-sized matrix. However, this computation is a little bit complicated for GPU computation. One voxel of the CT image may probably affect a number of pixels on the DRR image during the projection simulation, so that in the gradient computation, it is inevitable to consider it; that is, when mapping back the DRRs to the volume image, there is probably more than one ray line going through one voxel.
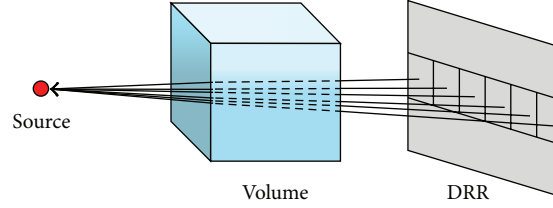
FIGURE 4: Subdividing the DRR image into small subgroups for GPU parallel computation.

If the original ray tracing implementation is directly used to parallelize all ray lines in the gradient computation, there will be a memory conflict within the GPU; that is, the gradient of the similarity energy E_{sim} with respect to one voxel intensity is updated by different ray lines simultaneously. To overcome this problem, the DRR image is subdivided into small subgroups so that the ray lines in different groups cannot go through the same voxel concurrently as shown in Figure 4. Currently, the computation can be computed on each voxel simultaneously by assigning it to each GPU thread. After that the total gradient of the similarity term as (16) is obtained by summing up all the gradient values from different subgroups and projections.

To maximally utilize the GPU's parallel computing power, the best subgroup size for the XCAT data is 8×8 pixels, and the head and neck (H&N) patient data is 16×24 pixels, which is determined during the preprocessing step. The general idea is computed based on similar triangle property according to the voxel scale, the pixel scale, the distance from source to volume position, and the distance from source to DRR position.

(b) Gradient of voxel intensity with respect to displacement vector of vertex i, $(\partial U_{\text{new}}^{\beta}(\mathbf{D}, i)/\partial \mathbf{D}_i)$: this can be done in parallel based on each mesh vertex when the deformed volume is computed, which is independent of the projection computation.

The CPU-based serial implementation of mesh-based 3D-2D registration method on XCAT data ($256 \times 256 \times 132$) with 60 projections ($256 \times 256$) takes about 2.5 hours; after using the GPU-based parallel implementation, it takes about 3 minutes, which is about 50 times faster.

2.2.5. A Multiresolution Scheme.
The size of H&N patient data used in this study is relatively large. CT volume data size is $512 \times 512 \times 140$ and CBCT projection size is 1024×768. The reconstruction running time of CPU-based implementation on mesh-based 3D-2D registration method with 30 projections is about 12 hours, while the running time of GPU-based implementation is about 20 minutes, which is about 36 times faster than the CPU-based one. The multiresolution scheme is used to further improve the speed. In the experiment, both the CT volume image and CBCT images are downsampled into different resolution levels (three levels for experiments on H&N patient data), from the coarsest level (CT volume: $256 \times 256 \times 70$, CBCT projection: 256×192, and time per iteration: 2.08 seconds) to the higher level (CT volume: $512 \times 512 \times 140$, CBCT projection: 512×384, and time per iteration:

9.91 seconds) and finally to the full resolution level (CT volume: $512 \times 512 \times 140$, CBCT projection: 1024×768, and time per iteration: 40.44 seconds). By using this strategy, the volumetric image reconstruction can be accomplished in 6.9 minutes, including 30 iterations of coarsest level, 15 iterations of higher level, and 5 iterations of full resolution level (about 60 times faster than CPU-based serial implementation). It is comparable to the fastest iterative CBCT techniques.

2.3. A Boundary-Based 3D-2D DIR.
In order to further improve the computational speed of the proposed 3D-2D DIR method, in this section, we introduce a boundary-based 3D-2D DIR to obtain a good initial deformation; then the feature-based nonuniform meshing for 3D-2D DIR method, as mentioned in Section 2.2, is used to generate the final volumetric images.

2.3.1. Extraction of 3D and 2D Boundaries.
After generating the feature-based nonuniform meshes (Section 2.1), both planning CT images and CBCT projections are preprocessed to create binary masks by setting "one" inside of the studied tissue and "zero" outside as shown in Figure 5(b). Then, the 3D tissue surface and CBCT projection boundaries are extracted by Canny edge detector [37] (Figure 5(c)).

2.3.2. Computation of Projections by Splatting Method.
In order to directly and conveniently control the updated positions of the deformed anatomy surface voxels, we prefer to use the splatting method [38] to generate projections of 3D surface, instead of using the ray tracing method. This is one main advantage of splatting method. Every voxel's contribution to a projection is mapped directly onto the image plane by a kernel centered on the voxel as shown in Figure 6. This reconstruction kernel is called a "splat" or "footprint":

$$
\begin{aligned}
f(x, y) &= \sum_{i,j,k} \text{voxel}(i, j, k) \, \text{kernel}(x, y) \\
&= \sum_{i,j,k} \text{voxel}(i, j, k) \, \frac{1}{\sqrt{2\pi\sigma^2}} e^{-((x-s)^2/2\sigma^2 + (y-t)^2/2\sigma^2)},
\end{aligned}
\tag{17}
$$

where $f(x, y)$ is the final pixel intensity of the projection image and $\text{voxel}(i, j, k)$ is the intensity of voxel at (i, j, k) position. $\text{kernel}(x, y)$ is the "footprint function" centered at (s, t). x and y are the Gaussian kernel area within radius 3σ on the projection image. σ is the Gaussian kernel width.

3D CT image   0-1 binary 3D image   3D surface voxel

CBCT projection   0-1 binary CBCT projection image   CBCT projection boundaries
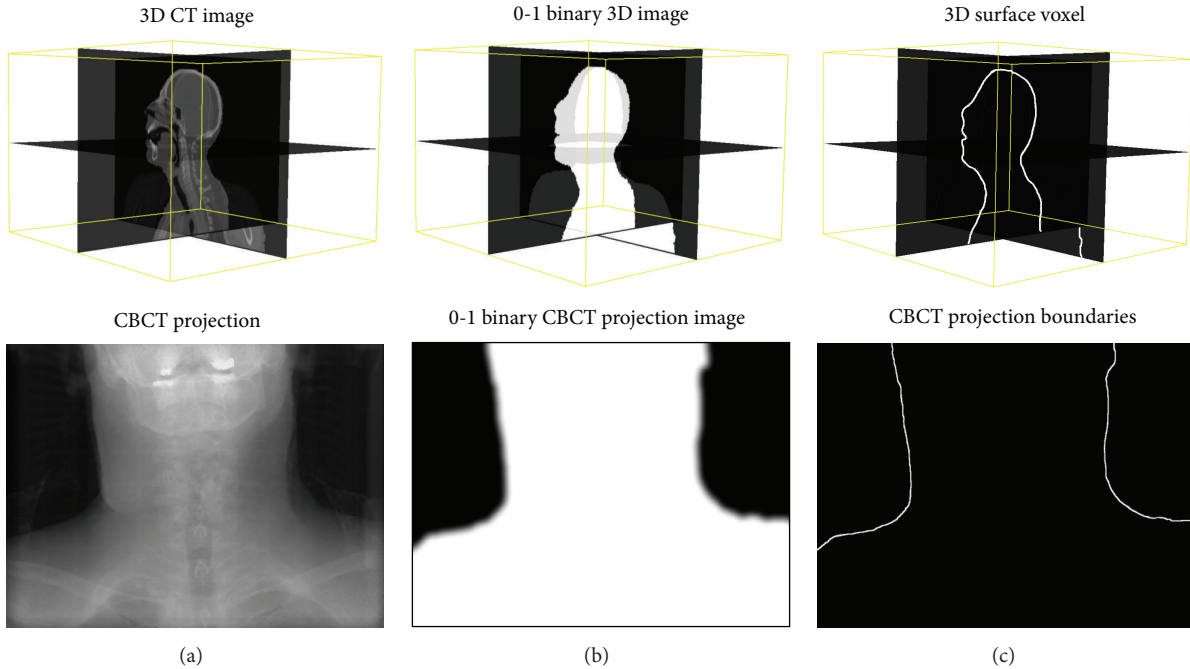
(a)   (b)   (c)

FIGURE 5: (a) Original images; (b) the 0-1 binary images; (c) boundaries of an H&N patient data.
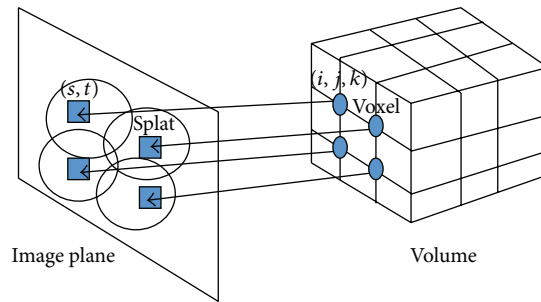

FIGURE 6: Splatting projection.

For perspective projection with antialiasing consideration, the Gaussian kernel radius should have dynamic sizes, which can be calculated from similar triangles shown in Figure 7 and

$$\frac{d1}{d2} = \frac{\text{voxel scale}}{r},$$

$$\sigma = 0.57r, \tag{18}$$

where $d1$ is the distance from the X-ray source to the voxel center and $d2$ is the distance from the source to kernel center on the image plane going through a specific voxel point. r is the size of splat, and voxel scale is the size of the voxel. The best coefficient value between σ and r is 0.57 based on our extensive experimental results. If the volume is a regular grid, voxel scale is fixed for all the voxels.

Another main advantage of splatting method over ray tracing is that splatting has a faster calculation speed; that is, it is very easy to ignore empty voxels (nonsurface voxels), which do not contribute to the final projection image. However, this is difficult to realize in ray tracing method.

It is noted that if we directly project the volume surface voxels (without considering kernels) onto the image plane, some pixels may be included there not belonging to the final 2D boundaries of the projection as shown at the top left of Figure 8. At the same time, in order to efficiently control the projection of the surface voxels, we do not use the projection with kernels to compute the boundaries. Instead, there are two projections computed at each gantry angle to extract the final 2D boundaries of the surface voxel projection. One is the projection with voxel kernels (similar to ordinary DRR computation) shown at the top right of Figure 8 and the other is without kernels (directly project each surface voxel onto the image plane) shown at the top left of Figure 8. We have to use the projections with kernels to compute rough DRRs (possibly aliasing exists, but we only care the image boundaries) and then employ it to filter out the exact 2D projection boundaries of the deformed anatomy surface shown at the bottom of Figure 9.
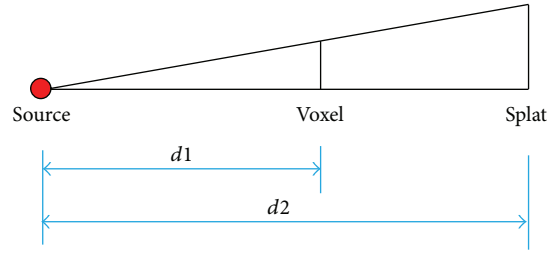
FIGURE 7: Geometry of the perspective projection with antialiasing for splatting method.
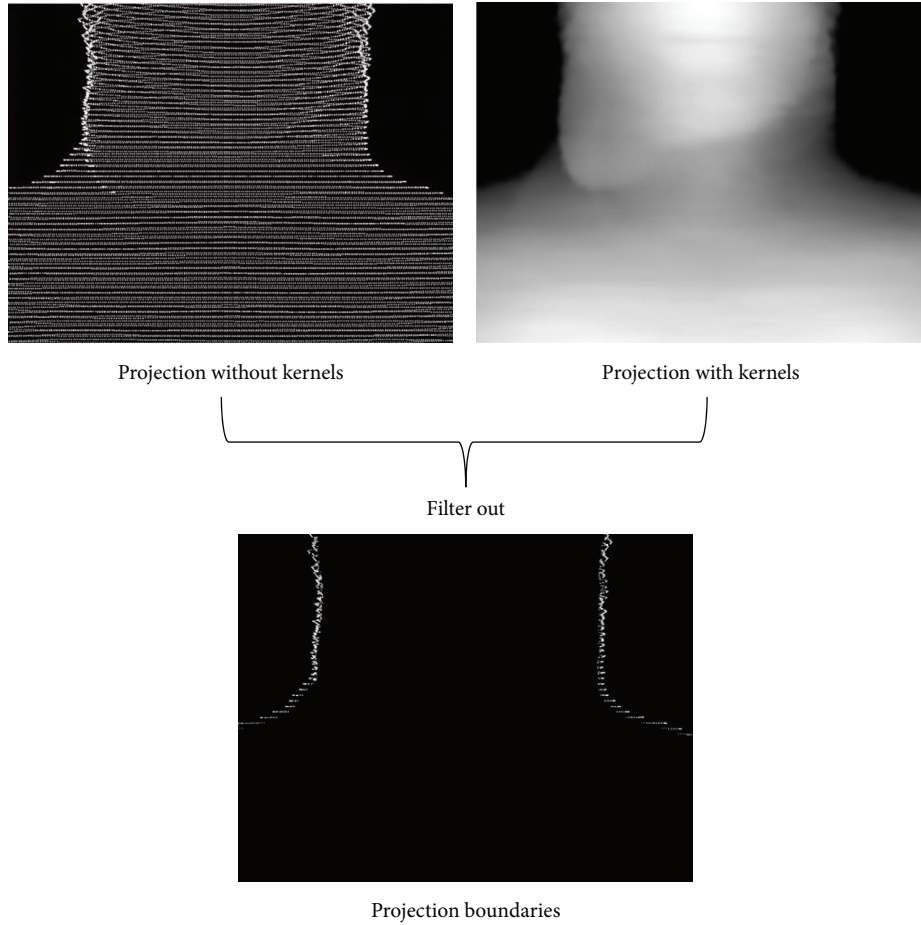

Projection without kernels

Projection with kernels

Filter out

Projection boundaries

FIGURE 8: Computation of 2D projection boundaries of the deformed anatomy surface from an H&N patient data.

2.3.3. Optimization of Boundary-Based 3D-2D DIR Energy. The computed projections of 3D surface are compared with corresponding 2D projection boundaries from CBCT scans, and the primary DVF is iteratively optimized to obtain a good initial deformation for final volumetric image. The surface deformation is optimized by minimizing the total energy E_{bound}, which includes two terms, the regularization used to achieve smoothness of the DVF and the similarity between the two images that is different from the previous intensity-based formulation. Here, the projections of the anatomy surface voxel are compared with corresponding 2D projection boundaries from CBCT scans:

$$
\begin{aligned}
E_{\text{bound}} \\
= \mu L\,(\mathbf{D}) \\
+ \sum_{m=1}^{N_P} \left\| \text{dist}_{\min} \left(R_{\text{bound}}\,(\mathbf{D}, \theta_m)\,, I_{\text{bound}}\,(\theta_m) \right) \right\|^2,
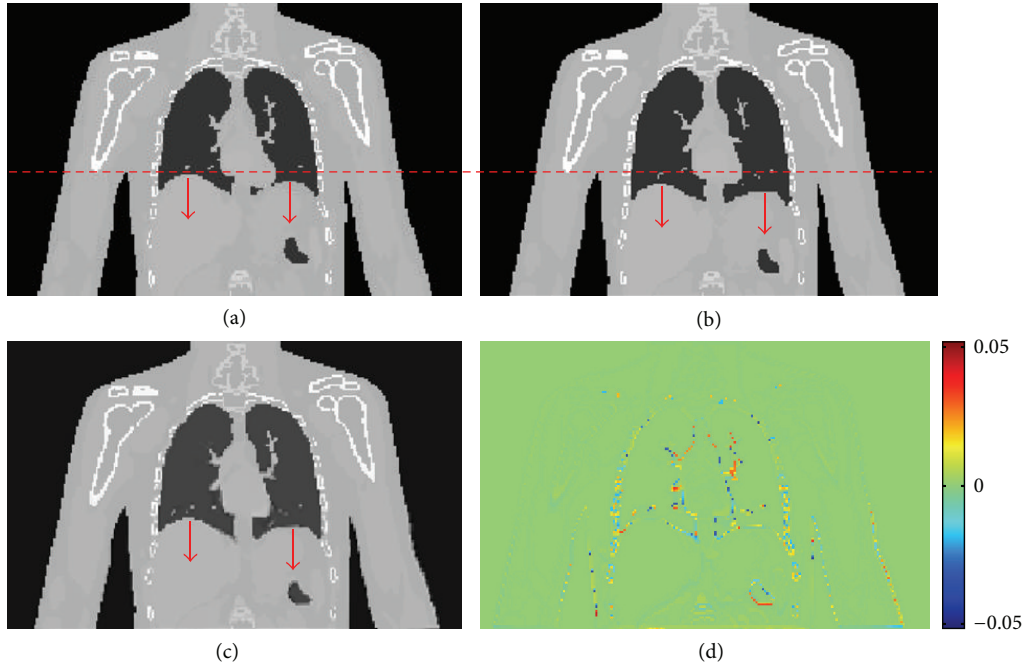\end{aligned}
\tag{19}
$$

Figure 9: Demonstration of XCAT male phantom results. (a) Original image (Phase 1); (b) target image (Phase 4); (c) deformed image from Phase 1; (d) differences between deformed and target images.

where μ is a weighting factor and empirically set at 10.0 for the experiments. $\mathbf{D}$ is the DVF defined on the mesh vertices. $L(\mathbf{D})$ is regularization term defined as in (10). The second term of the energy function indicates the similarity between the CBCT projections boundaries $I_{\text{bound}}(\theta_m)$ acquired at gantry angle θ_m beforehand and the projections $R_{\text{bound}}(\mathbf{D}, \theta_m)$ created after the deformation of $\mathbf{D}$ applied on planning CT body surface at the same gantry angle. N_p is the number of projections involved. This term is a summation of the square of the shortest Euclidean distance between every pixel on projections of the deformed CT surface and the corresponding CBCT projection boundaries. k-d tree data structure is applied to efficiently search such nearest pixels for boundaries [38].

L-BFGS algorithm is used to optimize the DVF ($\mathbf{D}$). The gradient of the energy function E_{bound} with respect to $\mathbf{D}$ can be calculated as follows:

$$\nabla E_{\text{bound}} = \mu \nabla L(\mathbf{D})$$
$$+ 2 \sum_{m=1}^{N_p} \text{dist}_{\min}(R_{\text{bound}}(\mathbf{D}, \theta_m), I_{\text{bound}}(\theta_m)) \quad (20)$$
$$\nabla \text{dist}_{\min}(R_{\text{bound}}(\mathbf{D}, \theta_m), I_{\text{bound}}(\theta_m)).$$

$\nabla \text{dist}_{\min}(R_{\text{bound}}(\mathbf{D}, \theta_m), I_{\text{bound}}(\theta_m))$ can be computed numerically by finite difference with a small $\Delta \mathbf{D}$.

Because we do not need the exact DRRs of the deformed 3D anatomy image, the resampling is not required. What we focus on is the updated voxel positions of the deformed 3D anatomy surface. Then we can use the splatting method to compute the projections of the deformed volume surface.

After the above boundary-based registration, the primary DVF is obtained and then applied in further complete intensity-based DIR as the initial deformation. As a result, the final volumetric images are obtained by applying the optimized DVF to planning CT images.

3. Results

The algorithms are implemented by using Microsoft Visual C++ 2010, MATLAB R2013a, and NVIDIA CUDA 5.5. For the hardware platform, the experiments are run on a desktop computer with Intel® Xeon E5645 CPU with 2.40 GHz, 34 GB DDR3 RAM, and NVIDIA GeForce GTX 780 Ti GPU with 3 GB GDDR5 video memory.

We evaluate and compare our proposed nonuniform tetrahedral meshing for 3D-2D DIR with other 3D-2D DIR methods, that is, voxel-based method [1–3], uniform orthogonal grid mesh [16], and uniform tetrahedral mesh on image visualization and quantitative evaluations on two XCAT phantoms and five H&N cancer patients.

3.1. Evaluation. This method is evaluated thoroughly by using two sets of digital XCAT phantoms and H&N patient data. Taking the XCAT male phantom data for example, two sets of 3D images, representing the same patient (phantom) at two different respiratory phases, are created. Both the beating heart and respiratory motions are considered, and in order to simulate the large deformation, the max diaphragm motion is set to 10 cm. Phase 1 and Phase 4 are shown in Figures 9(a) and 9(b). Phase 1 data is used as the original planning CT image, while Phase 4 data represents daily CBCT images.

(a)   (b)   (c)
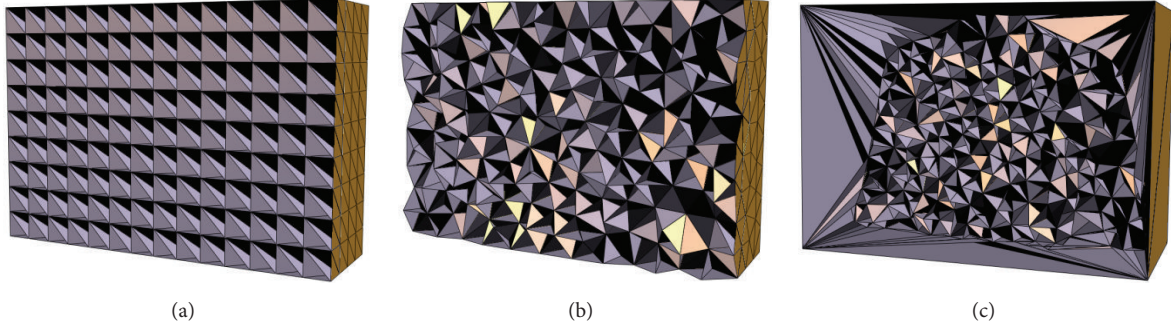
FIGURE 10: Three types of meshing for XCAT male phantom data. (a) A uniform orthogonal grid; (b) a uniform tetrahedron mesh; (c) a feature-based nonuniform tetrahedron mesh.

The deformation vector of every voxel between these two phases is provided by the XCAT software and it is used as the ground truth of DVF for evaluation. A set of DRRs from Phase 4 data is created using a ray tracing method with noise simulation as mentioned in Section 2.2.2 and subsequently used as the raw projections from the daily CBCT. The original CT (Phase 1 data) is deformed to fit the raw projections of daily CBCT and finally a new set of volumetric images is created and compared with the ground truth, Phase 4 data. Simultaneously, the final DVF is compared with the DVF obtained from the XCAT software.

A conventional normalized cross correlation (NCC) is used to evaluate the similarity (i.e., the linear correlation) of 3D images and DVFs:

$$\text{NCC} = \frac{\sum_{i=1}^{N} \left[F_{\text{intp}}(i) - \overline{F}_{\text{intp}} \right] \left[F(i) - \overline{F} \right]}{\sqrt{\sum_{i=1}^{N} \left[F_{\text{intp}}(i) - \overline{F}_{\text{intp}} \right]^2} \sqrt{\sum_{i=1}^{N} \left[F(i) - \overline{F} \right]^2}}, \quad (21)$$

where $F_{\text{intp}}(i)$ and $F(i)$ are the interpolated and target values, respectively, over N voxels. $\overline{F}_{\text{intp}}$ and $\overline{F}$ are the average values of the interpolated and target values. The range of the NCC is $[-1, 1]$. If NCC is 1, it means two values are exactly the same. The larger the NCC is, the more similar the values are.

The normalized root mean square error (NRMSE) between the interpolated values $F_{\text{intp}}(i)$ and the target values $F(i)$ is also used for comparison of 3D images and DVFs and it denotes the related error:

$$\text{NRMSE} = \sqrt{\frac{\sum_{i=1}^{N} \left[F_{\text{intp}}(i) - F(i) \right]^2}{\sum_{i=1}^{N} \left[F(i) \right]^2}}. \quad (22)$$

The range of the NRMSE is $[0, +\infty)$. If NRMSE is 0, it means two values are exactly the same. The smaller the NRMSE is, the more similar the values are.

To demonstrate the advantage of using feature-based nonuniform mesh, results of three types of meshing methods are compared. These are (1) a bounding box uniform orthogonal grid in Figure 10(a), as mentioned in [9]; (2) a uniform tetrahedron mesh in Figure 10(b); (3) a feature-based nonuniform tetrahedron mesh in Figure 10(c). A voxel-based deformation method is also evaluated.

3.2. Meshing Computation. In order to apply the meshing-based method in the 3D-2D image registration framework, we have to compute the meshes at first. The number of vertices of the tetrahedron meshes in XCAT phantom data and H&N patient data are all around 1,000; hence the execution time of the particle-based meshing method is the same. H&N patient data ($512 \times 512 \times 140$) is larger than XCAT data ($256 \times 256 \times 132$), so it takes more time in the preprocessing steps of image feature edges computation and the density field computation for feature-based tetrahedron mesh generation. Compared with the uniform orthogonal grid mesh generation (5 seconds), the isotropic tetrahedron mesh generation needs 10 seconds, and the feature-based tetrahedron mesh generation needs more time in preprocessing steps: (a) compute the image feature edges: 3.5 seconds (XCAT data) versus 15 seconds (H&N patient data); (b) compute the density field: 0.4 minutes (XCAT data) versus 1.5 minutes (H&N patient data); (c) run particle-based meshing framework: 10 seconds. Uniform and feature-based nonuniform tetrahedron meshes can be generated by our particle-based meshing approach only if the desired density field is available (the density field of uniform tetrahedron mesh is globally uniform). Once these meshes are generated, they are used by the 3D-2D DIR framework, and no additional computation is required for the meshes. It should be noted that the mesh generation could be done in advance as soon as the planning CT is performed. So the time for this preprocess can be hidden for the image registration process.

3.3. XCAT Phantom Data. Feature-based nonuniform tetrahedral meshes are created using approximately 1,000 vertices on the original CT image (Figure 9(a), Phase 1 of the XCAT male model). 60 DRRs are created from the target image (Figure 9(b), Phase 4) at 60 different gantry angles equally spaced over 360 degrees. The new volumetric images are obtained by optimizing the deformation of the meshes by comparing the 60 DRRs of the deformed images with the corresponding projections of the target images in 100 iterations of the elastic registration algorithm. Figure 9(c) shows that the new reconstructed images are very close to the target images. Their differences are illustrated in Figure 9(d), which are very small.
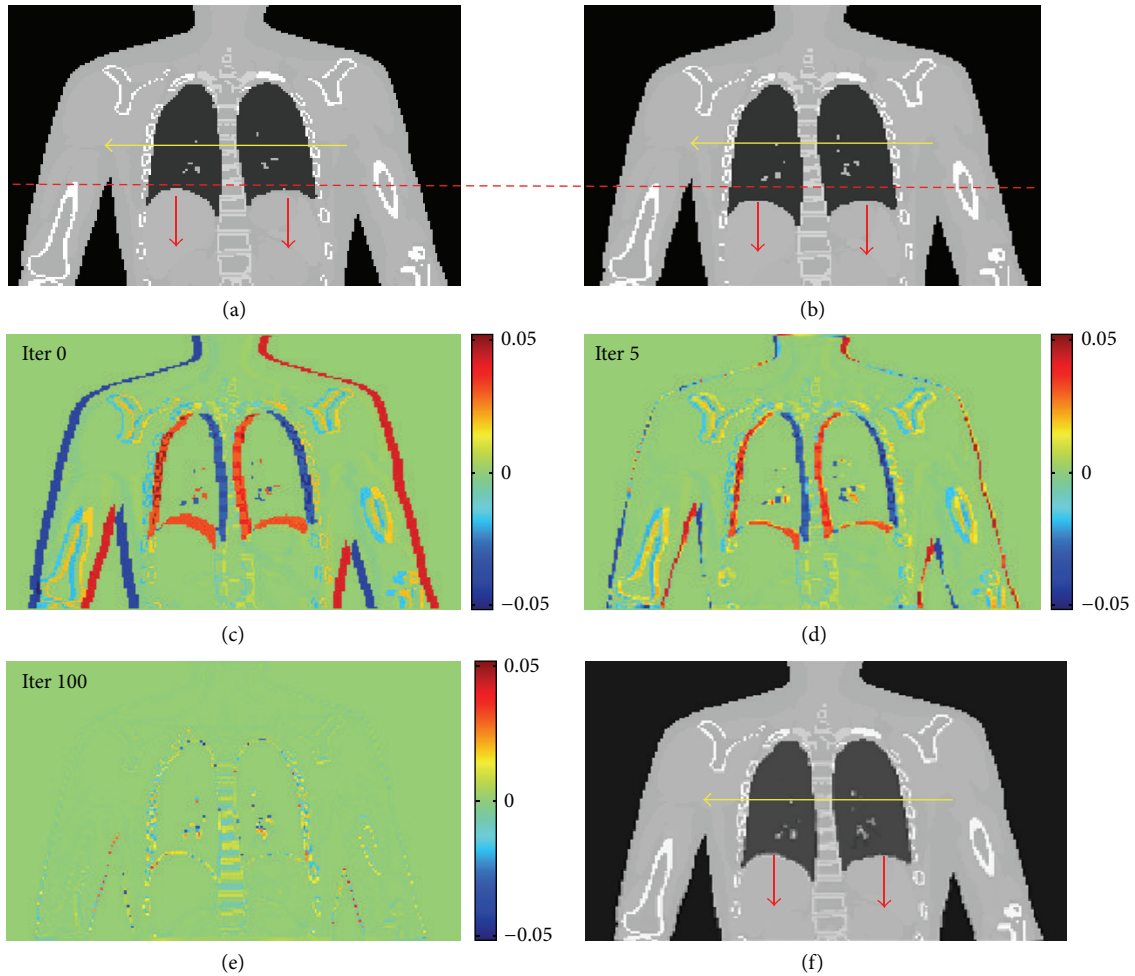
FIGURE 11: Demonstration of XCAT female phantom results. (a) Original image (Phase 1); (b) target image (Phase 4); (c) differences between deformed and target images at the beginning of optimization; (d) differences between deformed and target images at iteration 5; (e) differences between deformed and target images at iteration 100 (end of the optimization); (f) final deformed image from Phase 1.

To test the robustness and accuracy of the algorithm, another XCAT female phantom with more complicated motions including a large deformation between respiratory Phase 1 and Phase 4 and the translation (globally translated by four-voxel-size distance (i.e., about 4.68 mm) in the horizontal direction) is used. Other configurations are the same as the previous male phantom case. Figure 11 shows the intermediate and final results during the optimization process.

While a conventional CBCT reconstruction requires hundreds of projections, the mesh-based algorithm uses far fewer projections since the information from the planning CT image is used. The number of projections may vary from case to case. Table 1 lists the deformation results using varying numbers of projections. A larger number of projections do yield a higher NCC and lower NRMSE, though at the expense of longer calculation time and more radiation dose. A reasonable balance can be observed at 60 projections, as diminishing returns start to take effect by using a larger number of projections.

Comparisons of results from different meshing algorithms are shown in Tables 2 and 3. With regard to the similarity of the final images, the nonuniform mesh provides better results than the uniform orthogonal grid or uniform tetrahedron meshes in 100 optimization iterations, for the approximately same number of vertices. The uniform orthogonal grid mesh is the same as mentioned by Foteinos et al., who pointed out that it has the best registration performance in their experiments [16]. With more vertices, the nonuniform meshing method provides image results very close to the voxel-based method, which has as many vertices as voxels. It shows that the voxel-based deformation may yield good image intensity result, but the resulting DVF represents an unrealistic anatomical mapping. This is a drawback of voxel-based deformation and is due to its localized deformation (i.e., there are too many sampling points and it is easy to be trapped into some local minima). The feature-based nonuniform meshing method overcomes this drawback and yields more anatomically accurate DVF (both the NCC and NRMSE measurements on DVF of the nonuniform meshing

TABLE 1: Digital XCAT male phantom study results using various numbers of projections.

Number of projections used	10	20	30	60	90	120
NCC of images	0.9788	0.9812	0.9825	0.9855	0.9859	0.9860
NRMSE of images	0.1952	0.1836	0.1769	0.1612	0.1587	0.1585
NCC of DVF	0.7823	0.7946	0.8012	0.8150	0.8189	0.8201
NRMSE of DVF	1.0055	0.9076	0.8525	0.8118	0.7959	0.7951

Note. These comparison experiments are run through 100 iterations.

TABLE 2: Evaluation of reconstruction accuracy based on a digital XCAT male phantom.

	Uniform orthogonal grid	Uniform tetrahedron mesh	Nonuniform tetrahedron mesh	Nonuniform tetrahedron mesh	Voxel-based method
Number of vertices	1,050	987	1,005	10,004	8,650,762
NCC of images	0.9829	0.9835	0.9855	0.9858	0.9872
NRMSE of images	0.1749	0.1705	0.1612	0.1593	0.1514
NCC of DVF	0.7690	0.7829	0.8150	0.8265	0.6061
NRMSE of DVF	1.2366	1.0923	0.8118	0.8059	1.2484

Note. The three mesh-based methods are run through 100 iterations, while the voxel-based method needs 200 iterations.

TABLE 3: Evaluation of reconstruction accuracy based on a digital XCAT female phantom.

	Uniform orthogonal grid	Uniform tetrahedron mesh	Nonuniform tetrahedron mesh	Nonuniform tetrahedron mesh	Voxel-based method
Number of vertices	980	981	1,011	10,000	8,650,752
NCC of images	0.9789	0.9792	0.9829	0.9846	0.9775
NRMSE of images	0.1970	0.1954	0.1766	0.1682	0.2035
NCC of DVF	0.7687	0.7680	0.7672	0.7649	0.6292
NRMSE of DVF	1.3875	1.1358	0.9705	0.9777	1.0317

Note. The three mesh-based methods are run through 100 iterations, while the voxel-based method needs 200 iterations.

method are better than those of the voxel-based method, which were applied in [1–3]). Furthermore, from Figure 12, it is clearly seen that the energy curve of the feature-based meshing method (the red line with triangle marker) decreases dramatically faster than any other methods; that is, it has faster convergence speed during the image registration optimization. It is noted that because different meshes or voxel-based methods may have different regularization terms, such as different numbers of displacement vectors, we only compare the similarity term in (9) to make the comparison fair. With the large translation in the XCAT female phantom, the results (in Table 3) of voxel-based method are not as good as the meshed-based methods in both images and DVF measurements, due to its localized deformation and translation.

3.4. H&N Patient Data. This feature-based nonuniform meshing image registration method has been tested on five clinical data sets from the head and neck cancer patients H&N01~H&N05. Figure 13 illustrates the density field mapping based on image feature edges and feature-based tetrahedron mesh of H&N01 patient data. Figure 14 shows the deformation results from axial view (a big tumor on the right
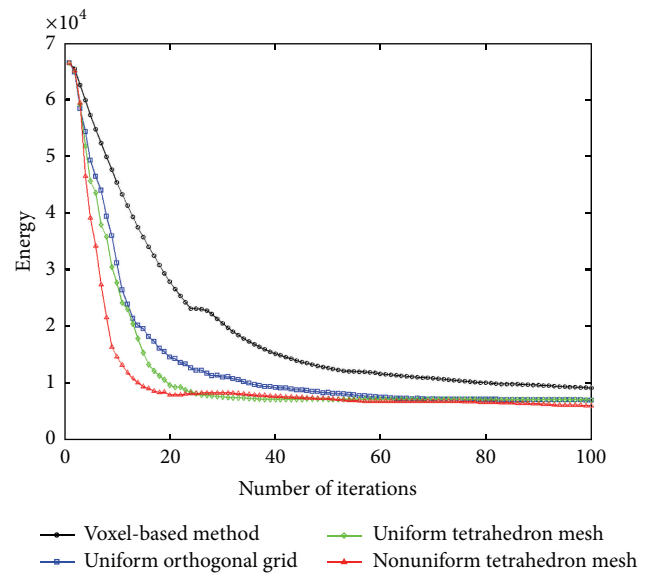

FIGURE 12: The similarity energy ($E_{\text{sim}}(\mathbf{D})$ term in (9)) curves of different methods (voxel-based, uniform orthogonal grid, uniform tetrahedron mesh, and nonuniform tetrahedron mesh) in image registration.

TABLE 4: Results with different numbers of projections on H&N01 patient data.

Number of projections used	10	20	30	60	90
NCC of images	0.8458	0.8459	0.8460	0.8460	0.8460
NRMSE of images	0.4798	0.4794	0.4792	0.4792	0.4792

Note. These comparison experiments are run through 50 iterations.

TABLE 5: Comparison of three meshes on the data of five head and neck cancer patients.

Patients	Uniform orthogonal grid	Uniform tetrahedron mesh	Nonuniform tetrahedron mesh
H&N01			
Number of vertices	936	992	1,007
NCC	0.8327	0.8358	0.8460
NRMSE	0.4988	0.4938	0.4792
H&N02			
Number of vertices	1,040	990	1,000
NCC	0.9036	0.9122	0.9134
NRMSE	0.4182	0.4116	0.4084
H&N03			
Number of vertices	980	1,000	998
NCC	0.8470	0.8471	0.8482
NRMSE	0.4748	0.4743	0.4722
H&N04			
Number of vertices	1,001	1,000	1,000
NCC	0.7756	0.7806	0.8111
NRMSE	0.5841	0.5817	0.5567
H&N05			
Number of vertices	980	1,000	995
NCC	0.7565	0.7690	0.7843
NRMSE	0.6278	0.6092	0.5712

Note. The three mesh-based methods are run through 50 iterations.

TABLE 6: Evaluation of boundary-based DIR accuracy on an H&N cancer patient data.

Status	Initial	Boundary-based DIR	Full DIR
NCC of images	0.7627	0.7875	0.7919
NRMSE of images	0.5938	0.5569	0.5479

side of the patient's chin shrinks) compared with the conventional CBCT reconstruction results of H&N01 patient data. The effects of various numbers of projections are also evaluated and the results are shown in Table 4. The three meshing methods, uniform orthogonal, uniform tetrahedron, and feature-based nonuniform tetrahedron (as shown in Figure 15), are evaluated with results shown in Table 5 for all patient data sets. The nonuniform meshing method again yields the highest accuracy and has faster convergence speed during the image registration. It is noted that, comparing the accuracies in the XCAT phantom data and the H&N patient data, the NRMSE seems larger in the patient cases (~0.5) than that in the XCAT cases (<0.2); this is because, in the patient case, the reconstructed CBCT image is computed based on the planning CT image, and the patient had the planning CT scan and daily CBCT scans on different days and machines, which may cause the differences in the image background, noises, and so forth, resulting in larger NRMSE

values. However, it is acceptable in the image results as shown in Figure 14.

3.5. Boundary-Based 3D-2D DIR Results. For the H&N05 cancer patient data, which has a large deformation on tumor during treatment, we use it to evaluate the effectiveness of the boundary-based 3D-2D DIR method. Figures 16(a) and 16(b) show the original and final differences of one projection boundary in boundary-based DIR. Figure 16(d) demonstrates that the final deformed CT images are very close to the target images (Figure 16(e)) and their differences are very small by further intensity-based 3D-2D DIR.

The accuracy of the boundary-based 3D-2D DIR and further intensity-based 3D-2D DIR are performed on H&N05 cancer patient data. Both the NCC and NRMSE shown in Table 6 demonstrate that boundary-based DIR can provide a good initial guess of the deformation for the further intensity-based 3D-2D DIR; that is, after the boundary-based 3D-2D
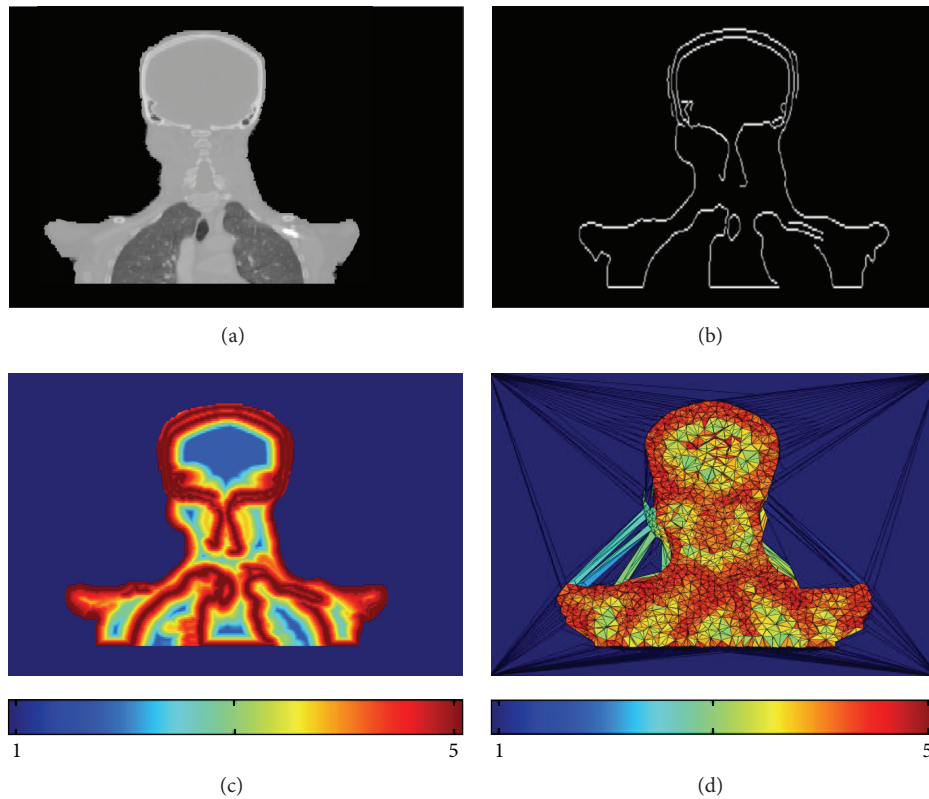
FIGURE 13: Demonstration of the feature-based mesh generation on the H&N01 patient data. (a) The original image; (b) extracted feature edges; (c) density field; (d) a 2D view of the interior meshes with color-mapping.
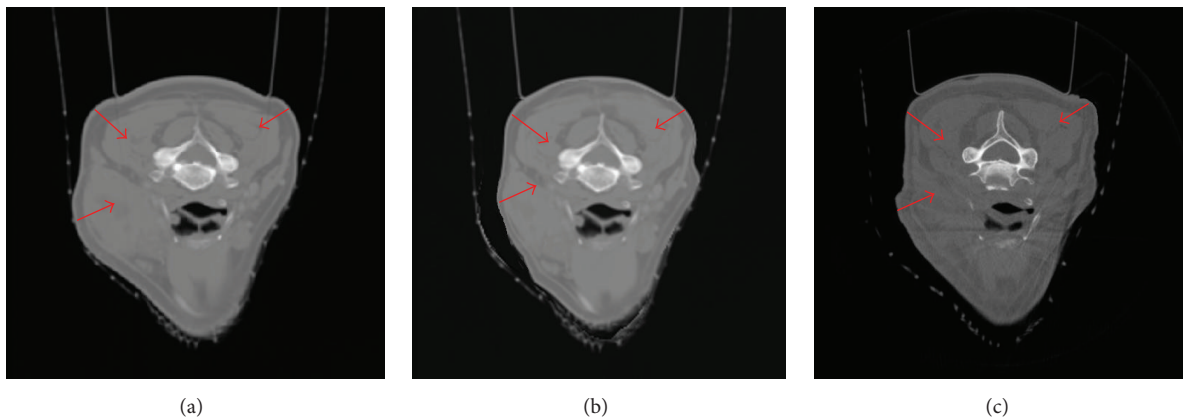

FIGURE 14: Demonstration of H&N01 patient results from axial view. (a) Original CT image; (b) deformed image; (c) target image (daily CBCT).

DIR, NCC and NRMSE are close to the final 3D-2D DIR accuracy results.

   With the GPU-based implementation, taking this H&N cancer patient data for example, our boundary-guided method takes 5.26 seconds per iteration, which is about 10 times faster than the non-boundary-guided method (i.e., 59.75 seconds). The multiresolution scheme is used to further improve the speed on both boundary-based 3D-2D DIR and

further intensity-based 3D-2D DIR. In the experiment, only the CBCT images are downsampled into different resolution levels (three levels for experiments on H&N05 patient data), from the coarsest level (CBCT projection: 256×192, time for boundary-based DIR: 1.37 seconds/iteration, and time for intensity-based DIR: 3.15 seconds/iteration) to the higher level (CBCT projection: 512×384, time for boundary-based DIR: 2.11 seconds/iteration, and time for intensity-based DIR:
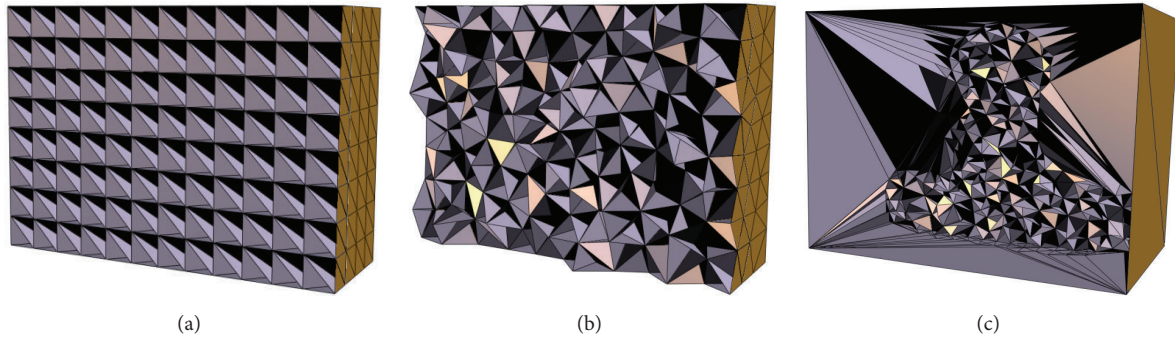
(a)　　　　　　　　　　(b)　　　　　　　　　　(c)

FIGURE 15: Three types of meshing for H&N01 cancer patient data. (a) A uniform orthogonal grid; (b) a uniform tetrahedron mesh; (c) a feature-based nonuniform tetrahedron mesh.

(a)　　　　　　　　　　　　　　　　　(b)
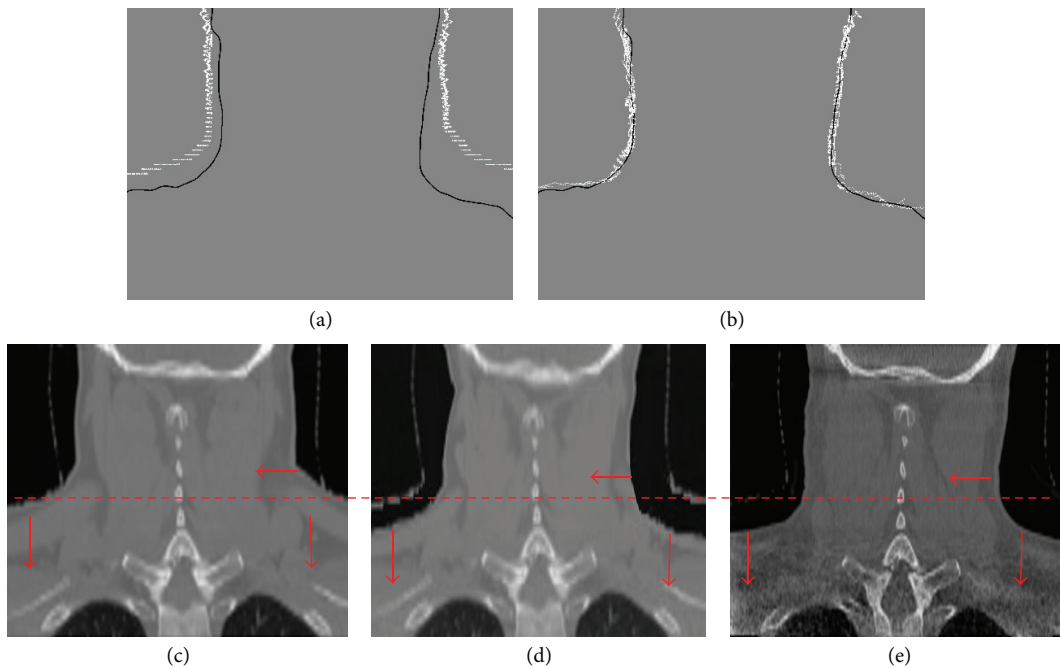
(c)　　　　　　　　　　(d)　　　　　　　　　　(e)

FIGURE 16: Demonstration of H&N05 patient data results. (a), (b) Differences between projection of the original/final surface voxel (white dots) and the target CBCT projection boundary (black dots); (c) original CT image; (d) reconstructed image from original CT image; (e) target image (daily CBCT).

15.01 seconds/iteration) and finally to the full resolution level (CBCT projection: 1024×768, time for boundary-based DIR: 5.26 seconds/iteration, and time for intensity-based DIR: 59.75 seconds/iteration). At the same time, the proposed method needs fewer optimization iterations (35 iterations compared with original 50 iterations). The significant advantage of this method is that, instead of registering the whole image, we just need to register the surface voxel and projection boundaries, which involves much fewer number of voxels and pixels. In addition, in the H&N patient data case, we only focus on the 3D surface and 2D projection boundaries, so that the effects of different image modalities will be ignored. By using both GPU implementation and multiresolution scheme, the volumetric image reconstruction of $512 \times 512 \times 140$ H&N cancer patient can be accomplished

within 3 minutes (compared with 6.9 minutes of the original 3D-2D DIR in Section 2.2.5), so that this boundary-based 3D-2D DIR method could be probably used in the clinically practical studies.

4. Conclusion and Future Work

The feature-based nonuniform meshing allows more sampling points to be placed in the important regions; thus the deformation can be controlled more precisely. With the approximately same numbers of sampling points (vertices), the feature-based nonuniform meshing method produces better registration results, where a larger NCC is obtained compared with the uniform orthogonal grid and the uniform tetrahedron meshes. While this improvement may seem

small, it is important to note that the NCC is very close to 1 because only minor anatomic changes occur. The NRMSE measurement is also provided to represent the differences between deformed images and the ground truth images. In H&N patient data, again, the feature-based nonuniform meshing method yields the highest accuracy of registration among the various methods.

It is intuitive that more sampling points (10,000 versus 1,000) lead to better results. In contrast, the voxel-based deformation provides the best image results in XCAT male phantom but requires using more than eight million sampling points. However, when the optimized DVF is compared with the ground truth DVF, the DVF of voxel-based deformation is significantly less similar to the ground truth than the feature-based nonuniform meshing method. Voxel-based deformation may yield better image intensity result, but the resulting DVF represents an unrealistic anatomical mapping. This is a drawback of voxel-based deformation and is due to its localized deformation. The feature-based meshing method overcomes this drawback and yields more anatomically accurate DVF.

As for the mesh quality, before deformation, the nonuniform tetrahedral meshes are generated based on the smooth density field as introduced in Section 2.1.2 so that we can obtain the high-quality adaptive tetrahedral meshes without any degradation and self-intersection. After image registration (deformation), the adaptive tetrahedral meshes are all good as well. These are two factors to guarantee no degraded and self-intersected tetrahedrons in the deformed meshes: (1) in the proposed energy function in (9), the regularization term $L(\mathbf{D})$ and weighting factor μ are to achieve the smoothness of the DVFs. (2) The target DVFs are quite smooth in both the XCAT phantoms and H&N cancer patients. Of course, if the DVFs are highly varying with sharp discontinuities or weighting factor μ is not set properly, the deformed tetrahedral meshes will have problems, such as degradations or self-intersections. Since the mesh quality study is beyond the scope of this work, future studies are warranted.

The repeated use of CBCT during a course of treatment could deliver high extra imaging dose to patients. For example, if weekly CBCT pelvis scans are performed with the conventional faction scheme, the total dose will be around 4.05 mSv/scan $\times$ 6 weeks = 24.3 mSV; and the total dose of head scans will be around 2.0 mSv/scan $\times$ 6 weeks = 12.0 mSV (Table II in [39]). If the daily CBCT is performed, the total dose will be much higher. Using this method, far fewer projections are needed to produce a set of high-quality volumetric deformed images than in conventional CBCT reconstruction, which can dramatically reduce the radiation dose during CBCT scans. Additionally, it is clearly seen that the feature-based nonuniform meshing method has faster convergence speed than other methods during the registration process.

Moreover, the proposed boundary-based 3D-2D DIR method can substantially further improve both the accuracy and the speed of reconstructing volumetric images by producing a good initial DVF. This eventually will lead to a fast and safe daily volumetric imaging with a very small number of projections for image-guided radiation therapy or online adaptive radiation therapy. There might be a limitation of this boundary-based method, if the deformation happens mainly in the internal organs. In the case of lung, its intensity is significantly different from that of chest wall, so we may segment lung and apply the proposed boundary-based 3D-2D DIR method only focusing on the lung first.

In the future, our feature-based nonuniform meshing method may also be applied to 4D images registration. A CBCT scan acquires approximately 600 hundred projections in a full rotation, and if they are sorted into ten respiratory phases, the corresponding 4D simulation CT set can be used to generate a high-quality, full 4D CBCT image set without exposing the patient to additional imaging dose. Currently, the feature-based nonuniform meshing method has been employed to some head and neck cancer patient data and achieved excellent results. In the future, we will investigate and determine the clinical accuracy of the method based on more patient data and statistical analysis in some follow-up applications for other cancer cases: such as breast, lung, and prostate cancers.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

References

[1] Y. Long, J. A. Fessler, and J. M. Balter, "Accuracy estimation for projection-to-volume targeting during rotational therapy: a feasibility study," *Medical Physics*, vol. 37, no. 6, pp. 2480–2490, 2010.

[2] R. S. Brock, A. Docef, and M. J. Murphy, "Reconstruction of a cone-beam CT image via forward iterative projection matching," *Medical Physics*, vol. 37, no. 12, pp. 6212–6220, 2010.

[3] J. Wang and X. Gu, "High-quality four-dimensional cone-beam CT by deforming prior images," *Physics in Medicine and Biology*, vol. 58, no. 2, pp. 231–246, 2013.

[4] L. Ren, I. J. Chetty, J. Zhang et al., "Development and clinical evaluation of a three-dimensional cone-beam computed tomography estimation method using a deformation field map," *International Journal of Radiation Oncology Biology Physics*, vol. 82, no. 5, pp. 1584–1593, 2012.

[5] G. P. Penney, J. Weese, J. A. Little, P. Desmedt, D. L. G. Hill, and D. J. Hawkes, "A comparison of similarity measures for use in 2D-3D medical image registration," *IEEE Transactions on Medical Imaging*, vol. 17, no. 4, pp. 586–595, 1998.

[6] J. Kim, F.-F. Yin, Y. Zhao, and J. H. Kim, "Effects of x-ray and CT image enhancements on the robustness and accuracy of a rigid 3D/2D image registration," *Medical Physics*, vol. 32, no. 4, pp. 866–873, 2005.

[7] R. Munbodh, H. D. Tagare, Z. Chen et al., "2D-3D registration for prostate radiation therapy based on a statistical model of transmission images," *Medical Physics*, vol. 36, no. 10, pp. 4555–4568, 2009.

[8] J. Kybic and M. Unser, "Fast parametric elastic image registration," *IEEE Transactions on Image Processing*, vol. 12, no. 11, pp. 1427–1442, 2003.

[9] A. Docef, M. Murphy, P. Keall, J. Siebers, and J. Williamson, "Forward CT reconstruction from limited projection data," in *Proceedings of the 19th Conference on Computer-Assisted Radiology and Surgery*, pp. 104–108, June 2005.

[10] R. Zeng, J. A. Fesslr, and J. M. Balter, "Estimating 3-D respiratory motion from orbiting views by tomographic image registration," *IEEE Transactions on Medical Imaging*, vol. 26, no. 2, pp. 153–163, 2007.

[11] X. Jia, B. Dong, Y. Lou, and S. B. Jiang, "GPU-based iterative cone-beam CT reconstruction using tight frame regularization," *Physics in Medicine and Biology*, vol. 56, no. 13, pp. 3787–3807, 2011.

[12] M. Ferrant, S. K. Warfield, A. Nabavi, F. A. Jolesz, R. Kikinis, and B. Macq, "Registration of 3-D intraoperative MR images of the brain using a finite-element biomechanical model," *IEEE Transactions on Medical Imaging*, vol. 20, no. 12, pp. 1384–1397, 2001.

[13] O. Clatz, H. Delingette, I.-F. Talos et al., "Robust nonrigid registration to capture brain shift from intraoperative MRI," *IEEE Transactions on Medical Imaging*, vol. 24, no. 11, pp. 1417–1427, 2005.

[14] B. Ahn and J. Kim, "Measurement and characterization of soft tissue behavior with surface deformation and force response under large deformations," *Medical Image Analysis*, vol. 14, no. 2, pp. 138–148, 2010.

[15] Y. Hu, T. J. Carter, H. U. Ahmed et al., "Modelling prostate motion for data fusion during image-guided interventions," *IEEE Transactions on Medical Imaging*, vol. 30, no. 11, pp. 1887–1900, 2011.

[16] P. Foteinos, Y. Liu, A. Chernikov, and N. Chrisochoides, "An evaluation of tetrahedral mesh generation for nonrigid registration of brain MRI," in *Computational Biomechanics for Medicine*, pp. 131–142, Springer, 2011.

[17] E. Haber, S. Heldmann, and J. Modersitzki, "Adaptive mesh refinement for nonparametric image registration," *SIAM Journal on Scientific Computing*, vol. 30, no. 6, pp. 3012–3027, 2008.

[18] M. Fogtmann and R. Larsen, "Adaptive mesh generation for image registration and segmentation," in *Proceedings of the 20th IEEE International Conference on Image Processing (ICIP '13)*, pp. 757–760, IEEE, Melbourne, Australia, September 2013.

[19] J. Zhang, J. Wang, X. Wang, and D. Feng, "The adaptive FEM elastic model for medical image registration," *Physics in Medicine and Biology*, vol. 59, no. 1, pp. 97–118, 2014.

[20] J. G. Brankov, Y. Yang, and M. N. Wernick, "Content-adaptive 3D mesh modeling for representation of volumetric images," in *Proceedings of the International Conference on Image Processing*, vol. 3, pp. 849–852, IEEE, June 2002.

[21] J. G. Brankov, Y. Yang, and M. N. Wernick, "Tomographic image reconstruction based on a content-adaptive mesh model," *IEEE Transactions on Medical Imaging*, vol. 23, no. 2, pp. 202–212, 2004.

[22] W.-S. V. Shih, W.-C. Lin, and C.-T. Chen, "Contour-model-guided nonlinear deformation model for intersubject image registration," in *Medical Imaging: Image Processing*, vol. 3034 of *Proceedings of SPIE*, pp. 611–620, International Society for Optical Engineering, Newport Beach, Calif, USA, February 1997.

[23] W.-N. Lie and C.-H. Chuang, "Contour-based image registration with local deformations," *Optical Engineering*, vol. 42, no. 5, pp. 1405–1416, 2003.

[24] X. Gu, B. Dong, J. Wang et al., "A contour-guided deformable image registration algorithm for adaptive radiotherapy," *Physics in Medicine and Biology*, vol. 58, no. 6, pp. 1889–1901, 2013.

[25] Z. Zhong, X. Guo, W. Wang et al., "Particle-based anisotropic surface meshing," *ACM Transactions on Graphics (TOG): SIGGRAPH 2013 Conference Proceedings*, vol. 32, no. 4, article 99, 2013.

[26] J. Nash, "C^1-isometric imbeddings," *Annals of Mathematics*, vol. 60, pp. 383–396, 1954.

[27] N. H. Kuiper, "On C1-isometric embeddings I," *Proceedings of the Koninklijke Nederlandse Akademie vanWetenschappen Series A*, vol. 58, pp. 545–556, 1955.

[28] M. do Carmo, *Riemannian Geometry*, Birkhäauser, 1992.

[29] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*, Springer, New York, NY, USA, 2008.

[30] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, no. 3, pp. 503–528, 1989.

[31] W. P. Segars and B. M. W. Tsui, "MCAT to XCAT: the evolution of 4-D computerized phantoms for imaging research," *Proceedings of the IEEE*, vol. 97, no. 12, pp. 1954–1968, 2009.

[32] H. Coxeter, *Introduction to Geometry*, John Wiley & Sons, 2nd edition, 1969.

[33] R. L. Siddon, "Fast calculation of the exact radiological path for a three-dimensional CT array," *Medical Physics*, vol. 12, no. 2, pp. 252–255, 1985.

[34] J. Wang, T. Li, H. Lu, and Z. Liang, "Penalized weighted least-squares approach to sinogram noise reduction and image reconstruction for low-dose X-ray computed tomography," *IEEE Transactions on Medical Imaging*, vol. 25, no. 10, pp. 1272–1283, 2006.

[35] P. J. La Rivière and D. M. Billmire, "Reduction of noise-induced streak artifacts in X-ray computed tomography through spline-based penalized-likelihood sinogram smoothing," *IEEE Transactions on Medical Imaging*, vol. 24, no. 1, pp. 105–111, 2005.

[36] K. Zhou, J. Huang, J. Snyder et al., "Large mesh deformation using the volumetric graph laplacian," *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 496–503, 2005.

[37] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[38] L. Westover, "Footprint evaluation for volume rendering," in *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '90)*, pp. 367–376, ACM, Dallas, Tex, USA, August 1990.

[39] R. A. Hälg, J. Besserer, and U. Schneider, "Systematic measurements of whole-body imaging dose distributions in image-guided radiation therapy," *Medical Physics*, vol. 39, no. 12, pp. 7650–7661, 2012.