GPU-Based 4D Cone-Beam CT Reconstruction by Meshing Method

Zichun Zhong[1], Xuejun Gu[1], Weihua Mao[1], Xiaohu Guo[2] and Jing Wang[1]
[1]Department of Radiation Oncology, UT Southwestern Medical Center and [2]Department of Computer Science, UT Dallas

***Abstract*—Due to the limited number of projections at each phase, the image quality of a four-dimensional cone-beam CT (4D-CBCT) is often degraded, which decreases the accuracy of subsequent motion modeling. Various strategies have been proposed to enhance the image quality of 4D-CBCT. One of the promising methods is the simultaneous motion estimation and image reconstruction (SMEIR) approach. However, one drawback of SMEIR is the long computational time. The objective of this work is to enhance the computational speed of the SMEIR algorithm using feature-based tetrahedral meshing and GPU-based parallelization.**

***Index Terms*—4D-CBCT, motion-compensated image reconstruction, feature-based tetrahedral mesh, GPU**

I. Introduction

There are growing interests in using four-dimensional cone-beam computed tomography (4D-CBCT) for managing respiratory motion involves sites, such as lung and liver tumors, in radiation therapy. Among various reconstruction algorithms for 4D-CBCT, the simultaneous motion estimation and image reconstruction (SMEIR) approach has shown promising results [1]. The SMEIR algorithm performs model-based image reconstruction and obtains the updated motion model simultaneously. The SMEIR algorithm consists of two alternating steps: model-based image reconstruction and motion model estimation. The model-based image reconstruction can reconstruct a motion-compensated primary CBCT (m-pCBCT) at any phase by using the projections from all of the phases with explicit consideration of the deformable motion between different phases. Instead of utilizing the pre-determined motion model, the updated motion is obtained through matching the forward projection of the deformed m-pCBCT and measured projections of other phases of 4D-CBCT. In the original SMEIR algorithm, voxel-based deformation fields was employed, where large number of unknowns need to be estimated and the computational time is extremely long. In this work, we develop a feature-based mesh technique for SMEIR and a GPU-based parallel algorithm to further improve the efficiency and accuracy of the SMEIR approach.

Finite element method (FEM) is a numerical technique for finding approximate solutions to boundary value problems for differential equations. FEM is best understood from its practical application, for instance, mesh discretization of a continuous domain into a set of discrete sub-domains. FEM has been widely used in deformation estimation [2, 3]. Usually, FEM is exploited to achieve two important advantageous aspects: 1) to improve the efficiency of the deformation vector fields (DVF) estimation process due to a small number of sampling points compared to voxel-based sampling methods; and 2) to provide the smoothness of the DVF due to the smoothness constraint between elements and the interpolation within one element. The quality of the geometric discretization is crucial for the effectiveness of the deformation estimation applications. Generally speaking, according to the geometric information, meshes can be divided into two categories: surface mesh and volume mesh. For surface mesh methods [3-5], during the registration process, the object surface is directly tracked, but the accuracy of estimated deformation degrades for locations further away from the surface. For volume mesh methods [6-9], for instance, the uniform grid mesh is often employed for simplicity; however, the vertex positions and connectivities of this kind of mesh are irrelevant to image features or the places where the deformation mostly occurs. Therefore, it is necessary to generate meshes according to image features.

The goal of this work is to incorporate a feature-based meshing method into the SMEIR algorithm to improve its computational efficiency and motion estimation accuracy. When iteratively reconstructing a sequence of 3D volumetric images, i.e. a 4D volumetric image, the number of sampling points is critical 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 DVF estimation 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 highly non-linear image intensity variation areas); while fewer sampling points are placed within homogeneous or linear image intensity variation regions. In this way, boundaries and other important features can be directly represented by the displacements of the sampling points or represented by smaller-sized tetrahedrons, rather than interpolating from a regular grid or a larger-sized tetrahedron in the volume mesh. As a result, the deformation can be controlled more precisely.

II. Methods

Fig. 1 illustrates a flow chart of the proposed mesh-based 4D CBCT reconstruction technique.

This paper mainly focuses on: feature-based mesh generation, motion-compensated CBCT image reconstruction, updated motion model estimation, and GPU-based parallel acceleration, which are discussed in the following sections.

A. Feature-based mesh generation

Fig. 2 illustrates the feature-based mesh generation on a set of torso images of a digital phantom 4D NCAT [10]. In the feature-based mesh generation, the first step is to design a density field to match the volume image features. Original images are pre-analyzed using a Laplacian operator (searching for zero crossings in the second derivative of the image to find edges) to extract features including contour edges and boundaries between organs and tissues, which are regions of highly non-linear intensity variation. When the feature edges of the volume image are obtained, a density field could be calculated without manual segmentation (Fig. 2 (b)).

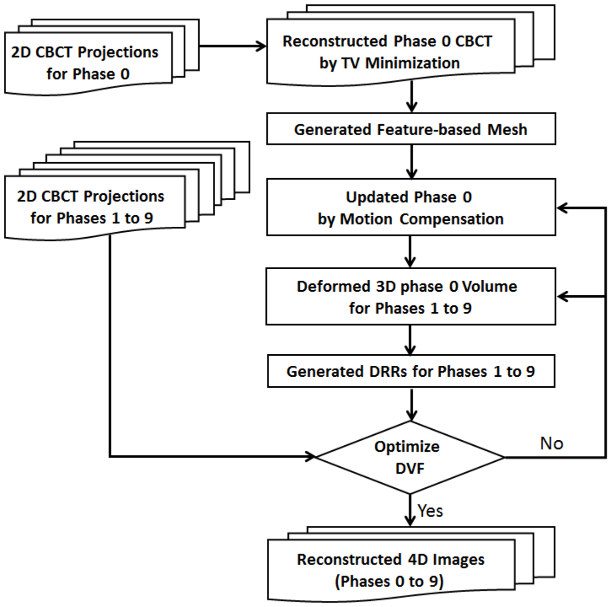
2D CBCT Projections for Phase 0
Reconstructed Phase 0 CBCT by TV Minimization
Generated Feature-based Mesh
2D CBCT Projections for Phases 1 to 9
Updated Phase 0 by Motion Compensation
Deformed 3D phase 0 Volume for Phases 1 to 9
Generated DRRs for Phases 1 to 9
Optimize DVF
No
Yes
Reconstructed 4D Images (Phases 0 to 9)

Fig.1 Flow chart of the proposed mesh-based 4D CBCT reconstruction method.

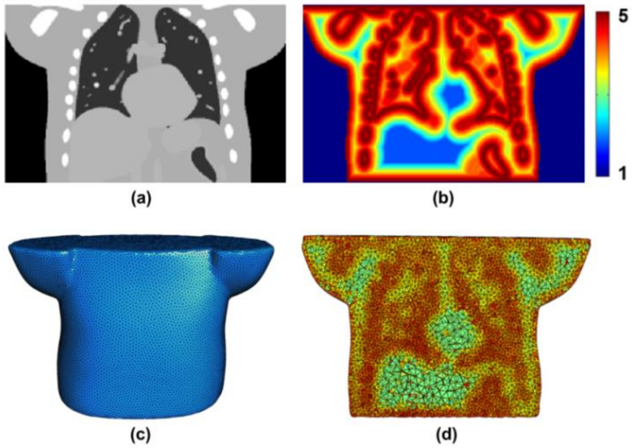

Fig.2 Demonstration of the feature-based mesh generation on a digital 4D NCAT phantom. (a) The original image at phase 0; (b) density field based on extracted feature edges; (c) boundary meshes well preserved the body surface; (d) a 2D view of the interior meshes with color-mapping.

After designing the density field, a binary image is 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. Then, the mesh vertices are automatically computed by particle-based meshing method [11]. The regions of highly non-linear intensity variation are densely positioned by vertices, while regions of constant or linear intensity variation are assigned fewer vertices. Following this process, vertex locations are conforming to the density field as illustrated in Fig. 2(b). Then volume meshes (tetrahedrons) are created based on the Delaunay triangulation of the vertices. More details about the algorithm of this mesh generation are mentioned in [11]. As a result, meshes corresponding to boundaries between organs and tissues are denser. The color-coded tetrahedrons of the generated feature-based mesh in Fig. 2 (c), (d) illustrate that the tetrahedral volumes are well conforming to the desired density defined by the features of the given image as well as the body surface. Since most deformations occur around the boundaries between organs and tissues, edges of contours, it is desirable to place more vertices (or sampling points) at features, while fewer vertices in non-feature 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-sized 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 and smoothly via interpolation as follows:

$$\mathbf{D} = \mathrm{INT}(\mathbf{W}), \quad (1)$$

where $\mathbf{W}$ denotes the DVF specified on mesh vertices, and $\mathbf{D}$ denotes the DVF specified on voxels. INT denotes the interpolation calculation.

B. *Motion-compensated CBCT image reconstruction*

The SMEIR algorithm consists of two alternating steps: motion-based image reconstruction and motion model estimation. After meshing generation, we use a motion-compensated reconstruction method to obtain a reference phase 0 for 4D-CBCT [1].

Suppose there are totally T phases in a 4D-CBCT, $\mathbf{P} = \{\mathbf{p}^t \mid t = 0,\ldots,T-1\}$ denotes the projections for all phases, and $\boldsymbol{\mu} = \{\boldsymbol{\mu}^t \mid t = 0,\ldots,T-1\}$ denotes attenuation coefficients for all phases. The mathematical relationship between $\mathbf{p}^t$ and $\boldsymbol{\mu}^t$ can be represented as follows:

$$\mathbf{p}^t = A\boldsymbol{\mu}^t, \quad (2)$$

where A is the projection matrix. The element a_{ij} of the matrix A is the weight of voxel j in $\boldsymbol{\mu}^t$ contributed to the pixel i in $\mathbf{p}^t$, i.e. the intersection length of projection ray i with voxel j, calculated by the ray tracing method [12].

Once the reference phase 0 is known and a motion model is available, the other phases [1,…, T-1] of the 4D-CBCT can be described as:

$$\boldsymbol{\mu}^t = D^{0\to t}\boldsymbol{\mu}^0 = \mathrm{INT}(W^{0\to t})\boldsymbol{\mu}^0, \quad (3)$$

where $D^{0\to t}$ can be considered as a voxel-based motion model computed from the interpolation of the DVF specified by the mesh model, and here it is the deformation matrix to transform 4D-CBCT from phase 0 to any phase t, $t \in [1,\ldots, T\text{-}1]$.

Now Eq. (2) can be rewritten according to phase 0 of the 4D-CBCT as follows:

$$\mathbf{p}^t = AD^{0\to t}\boldsymbol{\mu}^0. \quad (4)$$

Eq. (4) shows that the projection measurements from all phases of a 4D-CBCT can be contributed to reconstruct the reference image at phase 0 of a 4D-CBCT, leading to an m-pCBCT. Other phases of 4D-CBCT can be computed according to Eq. (3) based on phase 0 with motion models. In this paper, we use a modified simultaneous algebraic reconstruction technique (MSART) [1] to reconstruct the reference image at phase 0. In the MSART algorithm, the voxel values are updated iteratively as:

$$\mu_j^{0,(k+1)} = \mu_j^{0,(k)} + \lambda \frac{\sum_{t,n} d_{j,n}^{t\to 0} \sum_i \left[a_{i,n} \frac{p_i^t - \sum_n a_{i,n} \mu_n^{t,(k)}}{\sum_{n=1}^{J} a_{i,n}} \right]}{\sum_{t,n} d_{j,n}^{t\to 0} \sum_i a_{i,n}}, \qquad (5)$$

$$\mu_n^{t,(k)} = \sum_j d_{j,n}^{0\to t} \mu_j^{0,(k)}, \qquad (6)$$

where k is the iterative step, and λ is the relaxation factor which is set as 1.0 in our experiments. Eq. (6) describes the forward deformation that transforms 4D-CBCT from phase 0 to phase t, and n is voxel index at phase t. In Eq. (6), $d_{j,n}^{0\to t}$ denotes the element of the deformation matrix $D^{0\to t}$. In Eq. (5), the second term defines the inverse deformation that deforms the error image determined by projections at phase t to update the 4D-CBCT at phase 0. $d_{j,n}^{t\to 0}$ denotes the element of the inverse deformation matrix $D^{t\to 0}$. The DVF results of demons registration [13, 14] between the total variation (TV) minimization [15, 16] reconstructed 4D-CBCT at other phases and phase 0 were used as DVF initials for m-pCBCT reconstruction. After MSART reconstruction, the TV of the reconstructed m-pCBCT is minimized by the standard steepest descent method [1] to suppress the reconstructed image noise.

C. Updated motion model estimation

In this framework, the updated motion model can be obtained by matching the 4D-CBCT projection images at each phase with the corresponding forward projections of the deformed m-pCBCT. 4D-CBCT at phase 0 ($\boldsymbol{\mu}^0$) is obtained by the motion-compensated CBCT reconstruction algorithm as described in the previous section. Due to the DVF is defined at mesh vertices as shown in Eq. (1), Eq. (4) can be rewritten as:

$$\mathbf{p}^t = A\boldsymbol{\mu}^0(\mathbf{x} + \mathrm{INT}(\mathbf{w}^{0\to t})), \qquad (7)$$

where $\mathbf{x}$ denotes the coordinate vector of 4D-CBCT at phase t. The DVF $\mathbf{w}^{0\to t}$ can be computed by minimizing the energy function f, which includes two terms: the data fidelity between the two images and the regularization used to achieve smoothness constraint of the DVF:

$$f(\mathbf{w}^{0\to t}) = \left\| \mathbf{p}^t - A\boldsymbol{\mu}^0(\mathbf{x} + \mathrm{INT}(\mathbf{w}^{0\to t})) \right\|_{l_2}^2 + \beta L(\mathbf{w}^{0\to t}), \qquad (8)$$

where β is a parameter that controls the tradeoff between the data fidelity and the smoothness constraint of the DVF. $L(\mathbf{w}^{0\to t})$ is the smoothness constraint term defined as:

$$L(\mathbf{w}^{0\to t}) = \sum_{a=1}^{N_v} \sum_{d=1}^{3} [(\frac{\sum_{b\in N(a)} (\mathbf{w}_d^{0\to t}(b) - \mathbf{w}_d^{0\to t}(a))}{|N(a)|})^2], \qquad (9)$$

where $L(\mathbf{w}^{0\to t})$ is a summation of the square of Graph Laplacian operations [17] on the DVF (including three components: d = 1, 2, 3) over every vertex except those on the mesh external borders. N_v is the total number of the mesh vertices. $N(a)$ is the set of one-ring neighboring vertices (b) of vertex a. $|N(a)|$ is the size of set $N(a)$.

The total energy function considering all T phases is:

$$f(\mathbf{W}) = \sum_{t=0}^{T-1} \left[\left\| \mathbf{p}^t - A\boldsymbol{\mu}^0(\mathbf{x} + \mathrm{INT}(\mathbf{w}^{0\to t})) \right\|_{l_2}^2 + \beta L(\mathbf{w}^{0\to t}) \right], \qquad (10)$$

where $\mathbf{W} = \{\mathbf{w}^{0\to t} \mid t = 0, ..., T-1\}$. L-BFGS [18] (a quasi-Newton algorithm) optimization method is used to minimize f and obtain the optimized DVF $\mathbf{W}$. For each iteration of L-BFGS optimization, the energy f and its gradient $\frac{\partial f}{\partial \mathbf{W}}$ are updated. The parameter β is set at 1.0 in the digital phantom study in this paper. Ten total iterations are performed during the interleaved optimization after each MSART and TV minimization step.

D. GPU-based parallel acceleration

The entire process of this 4D-CBCT reconstruction method is implemented on GPU. The GPU card used in our experiments is an NVIDIA Tesla K40c with 12GB GDDR5 video memory. It has 2,880 CUDA cores utilizing such a GPU card with tremendous parallel computing ability that can significantly increase the computational efficiency. There are two time-consuming processes during the reconstruction. One is the projection generation in the total energy computation, and the other is the gradient computation of the total energy. It is straightforward to accomplish the ray tracing algorithm in parallel computation. For example, each pixel intensity of the projection is determined by accumulating all of the weighted voxel intensities through which one ray goes, i.e. integral of the intersection lengths of one projection ray going through voxels. This computation process is highly independent between each ray line. In this case, different GPU threads can compute each ray line simultaneously without conflict. Furthermore, because the projections at different phases can be all independently computed at the same time, in this case, the computation is fully parallelly calculated on GPU. The gradient computation of the total energy can be parallelly computed on GPU with the similar idea.

III. RESULTS

A 4D NCAT digital phantom was used to test the proposed mesh-based image reconstruction algorithm. Feature-based tetrahedral mesh is created on about 1,000 vertices based on the TV minimization result of 4D-CBCT at phase 0. In this study, ten breathing phases of the NCAT phantom were generated with a breathing period of 4 s. The maximum diaphragm motion is 20 mm and the maximum chest anterior–posterior motion is 12 mm during the respiration. A spherical 3D tumor with diameter of 10 mm was also simulated and is indicated by an arrow as shown in Fig. 3. The dimensions of the phantom were 256 × 256 × 150 with voxel size of 1 × 1 × 1 mm^3. Projections of ten phases of 4D-CBCT were generated using ray-tracing algorithm [12]. The dimensions of each projection were 300 × 200 with detector pixel size of 1 × 1 mm^2. 20

projections with noise model [19] were simulated for each phase of 4D-CBCT and were evenly distributed over 360°.

Fig. 3 shows images of the 4D NCAT phantom at the end-expiration phase. Fig. 3 (a) is the target image at phase 4, and (b) is the reconstructed image results by the proposed method; (c) shows the horizontal profiles through the center of coronal view images, which further illustrate the performance of our image reconstruction strategy. Not only are view aliasing artifacts suppressed, but the edges are also well-preserved for both bone structures and inside of the lung. Furthermore, from the image and profile, it can be clearly observed that the tumor position can be well captured as well.

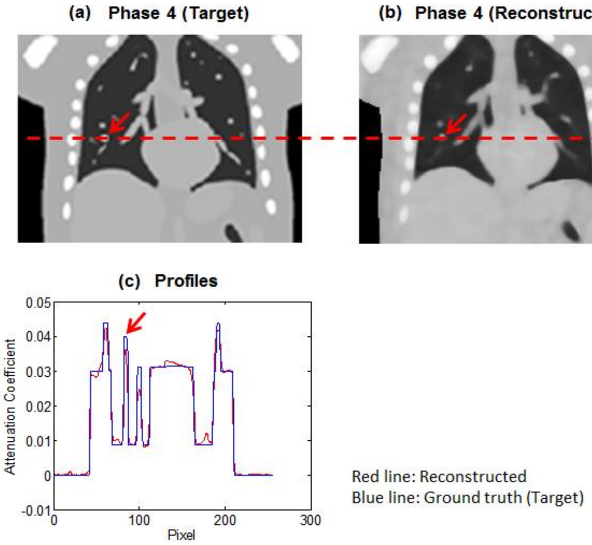

Fig. 3 Demonstration of 4D NCAT results. (a) Target image (Phase 4); (b) reconstructed image by the proposed method; (c) horizontal profiles through the center of coronal view images of (indicated by a read dashed line). The arrow indicates the tumor position.

TABLE 1 shows the timing information in each step per iteration during the optimization in the proposed GPU-based 4D CBCT reconstruction method. It is noted that the meshing generation step does not show in the table, since it only needs to compute once before motion compensated image reconstruction and DVF optimization iterations. In our phantom data, it needs 30 s for the mesh generation.

TABLE 1 Timing information of one iteration for different steps during the optimization.

Step		Time (s)
(1) Motion compensation		14.43
(2) DVF estimation	(2.1) Total energy computation	1.84
	(2.2) Gradient of total energy computation	7.74

The optimization totally needs to be run with 10 iterations of motion compensation (outer loop) and 100 iterations of DVF estimation (inner loop), so the total computational time of reconstruction of a 4D 256 × 256 × 150 CBCT image with 10 phases is 1102.3 s (18.37 m). Compared with the previous voxel-based CPU algorithm of SMEIR, which needs about 48 hours, the proposed mesh-based 4D-CBCT reconstruction method has largely improved the efficiency in computation (i.e. about 157 times faster than the original SMEIR method).

IV. CONCLUSION

The GPU-based parallel 4D CBCT reconstruction method uses the feature-based mesh for estimating motion model and demonstrates equivalent image result with previous SMEIR approach, with significantly improved computational speed.

REFERENCES

[1] J. Wang and X. Gu, "Simultaneous motion estimation and image reconstruction (SMEIR) for 4D cone-beam CT," *Med Phys,* vol. 40, p. 101912, 2013.

[2] M. Ferrant, S. Warfield, A. Nabavi, F. Jolesz, and R. Kikinis, "Registration of 3D intraoperative MR images of the brain using a finite element biomechanical model," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2000, pp. 19–28.

[3] O. Clatz, H. Delingette, I. Talos, A. Golby, R. Kikinis, F. Jolesz*, et al.*, "Robust nonrigid registration to capture brain shift from intraoperative MRI," *IEEE Trans. Med. Imaging,* vol. 24, 2005.

[4] B. Ahn and J. Kim, "Measurement and characterization of soft tissue behavior with surface deformation and force response under large deformations," *Med Image Anal,* vol. 14, pp. 138-48, Apr 2010.

[5] Y. Hu, T. J. Carter, H. U. Ahmed, M. Emberton, C. Allen, D. J. Hawkes, *et al.*, "Modelling prostate motion for data fusion during image-guided interventions," *IEEE Trans Med Imaging,* vol. 30, pp. 1887-900, Nov 2011.

[6] P. Foteinos, Y. Liu, A. Chernikov, and N. Chrisochoides, "An evaluation of tetrahedral mesh generation for nonrigid registration of brain MRI," *Computational Biomechanics for Medicine,* pp. 131–42, 2011.

[7] E. Haber, S. Heldmann, and J. Modersitzki, "Adaptive mesh refinement for nonparametric image registration," *Journal SIAM Journal on Scientific Computing,* vol. 30, pp. 3012–27, 2008.

[8] M. Fogtmann and R. Larsen, "Adaptive mesh generation for image registration and segmentation," presented at the 20th IEEE International Conference on Image Processing (ICIP), 2013.

[9] J. Zhang, J. Wang, X. Wang, and D. Feng, "The adaptive FEM elastic model for medical image registration," *Phys. Med. Biol.,* vol. 59, pp. 97–118, 2014.

[10] W. P. Segars, "Development and application of the new dynamic NURBS-based Cardiac-Torso (NCAT) phantom," *Ph.D. dissertation, University of North Carolina,* 2001.

[11] Z. Zhong, X. Guo, W. Wang, B. Lévy, F. Sun, Y. Liu*, et al.*, "Particle-based anisotropic surface meshing," *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2013),* vol. 32, p. 14, 2013.

[12] B. D. Man and S. Basu, "Distance-driven projection and backprojection in three dimensions," *Phys. Med. Biol.* , vol. 49, pp. 2463-75, 2004.

[13] J. P. Thirion, "Image matching as a diffusion process: An analogy with Maxwell's demons," *Med. Image Anal.* , vol. 2, pp. 243–60, 1998.

[14] X. Gu, H. Pan, Y. Liang, R. Castillo, D. Yang, D. Choi, *et al.*, "Implementation and evaluation of various demons deformable image registration algorithms on a GPU," *Phys. Med. Biol.* , vol. 55, pp. 207–19, 2010.

[15] T. Solberg, J. Wang, W. Mao, X. Zhang, and L. Xing, "Enhancement of 4D cone-beam computed tomography through constraint optimization," in *International Conference on the Use of Computers in Radiation Therapy*, Amsterdam, Netherlands, 2010.

[16] J. Song, Q. H. Liu, G. A. Johnson, and C. T. Badea, "Sparseness prior based iterative image reconstruction for retrospectively gated cardiac micro-CT," *Med. Phys.* , vol. 34, pp. 4476-83, 2007.

[17] K. Zhou, J. Huang, J. Snyder, X. G. Liu, H. J. Bao, B. N. Guo*, et al.*, "Large mesh deformation using the volumetric graph laplacian," *ACM Transactions on Graphics, (Proceedings of SIGGRAPH 2005),* vol. 24, pp. 496–503, 2005.

[18] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical Programming,* vol. 45, pp. 503 –28, 1989.

[19] J. Wang, T. Li, H. B. Lu, and Z. R. Liang, "Penalized weighted least-squares approach to sinogram noise reduction and image reconstruction for low-dose x-ray computed tomography," *IEEE Trans. Med. Imaging* vol. 25, pp. 1272-83, 2006.