Direct Volume Manipulation for Visualizing Intraoperative Liver Resection Process

Megumi Nakao[1], Yuya Oda[2], Kojiro Taura[3] and Kotaro Minato[2]
[1]Graduate School of Informatics, Kyoto University
[2]Graduate School of Information Science, Nara Institute of Science and Technology
[3]Dept. of Hepato-Biliary-Pancreatic and Transplant Surgery, Kyoto University Hospital

Abstract - This paper introduces a new design and application for direct volume manipulation for visualizing the intraoperative liver resection process. So far, interactive volume deformation and resection have been independently handled due to the difficulty of representing elastic behaviour of volumetric objects. Our framework models global shape editing and discontinuous local deformation by merging proxy geometry encoding and displacement mapping. A local-frame-based elastic model is presented to allow stable editing of the liver shape including bending and twisting while preserving the volume. Several tests using clinical CT data have confirmed the developed software and interface can represent the intraoperative state of liver and produce local views of reference vascular structures, which provides a "road map of vessels" that are key features when approaching occluded tumors during surgery.

Keywords —Volume manipulation, shape editing, liver resection, computer aided surgery

1 INTRODUCTION

The main priorities of oncologic liver surgery are complete removal of tumors within an appropriate safety margin and preservation of a sufficient volume of liver tissue to maintain hepatic function. Preoperative planning of liver resection is based on ultrasound (US), computed tomography (CT) and magnetic resonance imaging (MRI), and provides visualization of liver anatomy [1][2][3]. A patient-specific resection path is planned by considering three-dimensional vascular structures, and setting some vessels such as branches as anatomical reference points to be followed.

During surgery, careful ablation is required to achieve accurate resection while checking the reference vascular structures partially revealed in the process of resection. However, the anatomical information based on preoperative CT data differs from the intraoperative situation because of the surgical manipulation and deformation of the liver during resection [4][5] (see Figure 1). This makes it difficult for surgeons to accurately trace the resection path as preoperatively planned on the measured CT volume. Missing the path has possibility of yielding risks, for example, different vessels that are not regarded as anatomical references can appear. Intraoperative ultrasound is used to detect the tumor in the deformed state as a guide for liver resection [6]. However, there is still difficulty in understanding the 3D relationship between the tumor and the vascular structures based on two-dimensional (2D) imaging of a small part of the liver.

Our motivation for this research was to overcome these difficulties, and we have designed a planning/navigation framework that represents the intraoperative state of liver and produce local views of reference vascular structures. Unlike conventional applications, we focus on sequential visualization of the intraoperative resection process in addition to planning of the global resection path. Since we aim to provide a computer aided design (CAD) tool for liver resection surgery, the software should be intuitive and user friendly for medical staff. Since surface models also run into limitations in the handling of internal structures and the complex resection process, providing direct volume manipulation for the generated rendered image is a key concept in our framework.

A number of recent studies have sought to represent volume deformation and editing using point sampling and 3D texture interpolation on GPUs [7][8][9][10]. Although few models allow physically strict descriptions, this approach has the potential to enable interactive manipulation while providing visually valid deformation. In the direct manipulation of the volume, we note that control-point-based approaches [11][12] may actually result in indirect manipulation. This is inevitable, as the user manipulates control points rather than volumetric objects. Moreover, various technical hurdles have yet to be cleared for various types of geometry-based manipulation such as the representation of the elastic behaviour of soft tissues, the arbitrary discontinuous deformation for volumetric cuts and the rendering of time-varying ablation procedure. Specifically, we focus on the importance of three technical points: the global shape editing of liver including large deformations, visualization of vascular structures partly appearing in the resection process and the design of a direct volume manipulation interface for end users.

As volume data are originally 3D scalar fields, they have no 3D geometry, surfaces, or physical information in explicit formats. In consideration of this property, we begin our paper by exploring how to model flexible 3D manipulations and the shape editing of volume data. We also describe the design of an interactive, direct manipulation environment and a volume rendering scheme to represent the resection process. Towards the end of the paper, we validate the performance of our techniques using case examples. In the context of visualization literature, this work contributes in the following points.

- It introduces a design and application of direct volume manipulation for visualizing the intraoperative liver resection process.
- Both global shape editing and discontinuous local deformation are modelled by merging proxy geometry representation and displacement mapping.
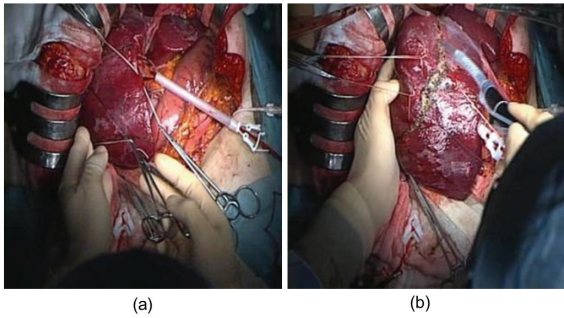- It presents a local-frame-based elastic model for editing

Figure 1 Snapshots of liver resection surgery, (a) intraoperative liver deformation and (b) ablation of the deformed liver

the liver shape. Our algorithm makes possible to represent twisting and bending states with stability while preserving the volume.

- Visualization algorithms to represent local vascular structures are described. Discontinuous deformation based on arbitrary resection path is modelled.
- A new design for a mouse-based intuitive interface for direct volume manipulation is introduced and demonstrated on clinical examples using patient's CT data.

2 RELATED WORK

A. Volume manipulation

The representation of volume deformation has been studied for at least a decade. The earlier studies focus on the construction of deformed volumes from initial states based on image morphing[13] or rigid transformation[14]. The various types of manipulation techniques have been explored by designing more complex transformations of the volume. More recently, in volume illustration and other fields of application, various types of deformation have been formulated in more detail to explore the complex internal structures of volume data [6][12][15]. One way to represent volume deformation is to pre-computationally prepare for a transformation of the volume. This framework is categorized as a form of non-interactive manipulation. The approach is still popular in related fields of research, such as transfer functions[16] and displacement maps[10][17].

Other studies, meanwhile, have been exploring interactive manipulation techniques together with advances in GPU technology and programmable shaders. Resk-Salama et al. began by proposing a basic framework for realizing interactive volume deformation[18][19]. In their framework, a 3D deformation is defined based on the transformation of proxy points placed in a segmented volume. The points are placed in a static and regular fashion before the manipulation begins. Masutani et al. have extended this approach in a finite element model (FEM) for simulating the elastic behaviour of organs [20]. George et al. presented an efficient scalable pipeline for GPU tetrahedral grid rendering[21]. Correa et al. recently presented an efficient design for volume deformation using scattered control points [11]. In their framework, the user manually places control points on cross sections of the volume and manipulates them to goal positions. The deformation is rendered via inverse displacement maps [17]. Although it succeeds in controlling some types of deformation, the method of manipulation is still

indirect, and independent of the structure of the volume. Furthermore, the small number of manually placed 2D control points restricts the degree-of-freedom of the deformation.

In spite of the earlier efforts heretofore described, several issues remain to be solved before direct volume manipulation becomes a reality. Specifically, the technical challenge is on-the-fly, direct handling of implicit geometry stored in volume data. The friendliest environment will allow the user to freely and directly manipulate the volume of interest while observing the rendered image. No method for direct volume manipulation has yet been formulated from this point of view. This paper introduces a set of direct volume manipulation techniques. Specifically, we extend point sampling approaches [11][19] to a more generalized format, that of the proxy volume mesh, and present a new framework for handling geometry and rendering for direct volume manipulation.

B. Large deformation

The finite element method (FEM) divides the whole elastic body into finite elements, and physically strict equations are solved by large-scale linear equations [22][23][24]. This method is characterized by the ability to directly use physical values such as Young's modulus and Poisson's ratio, and linear FEM makes it possible to quickly provide high precision deformation results in small deformations [25]. However, as the linear model is based on the Cauchy strain tensor theory, it produces unnatural deformation results due to increases in the volume during twisting, bending manipulation [26].

Non-linear finite element models [27][28][29] based on Green-Lagrange strain have also been studied in an effort to simulate large deformation. However, non-linear models increase the amount of calculation, making it difficult to achieve calculations in real time. Kikuchi et al improved the calculation method and reduced calculation costs, making it possible to conduct real-time deformation while preserving volume even in the case of large deformation [29]. There is also an approach in which the rotational components in an element are eliminated to evaluate deformation and rotation is applied after deformation in order to reduce calculation error [30][31]. In this way, representation of the large deformation of elastic bodies is being partially achieved. However, as these methods are based on a dynamic solution in which deformation is calculated by means of time integration, there is a possibility that the system will become unstable.

The mesh editing method [32][33] has been studied in the field of interactive computer graphics and computer-aided design (CAD). In this method, deformation is not modelled dynamically and evaluation functions that pre-serve shapes mathematically are defined and the minimization problem is solved in order to edit the shape of the model so as to fulfil the constraints. In order to achieve natural shape editing that includes rotation, Lipman et al proposed an approach based on the Cartan moving frame theory, in which local coordinates are placed at each vertex, and all vertex positions are represented on the local coordinates [34]. With this method, local rotation of an object can be modelled as rotation of the local coordinates, and natural shape interpolation is possible. Moreover, they touch on the preservation of volume [33]. As far as we surveyed, there are as yet no examples simulating elastic deformation based on this approach. However, this method is thought to be suitable even when dealing with the twisting and bending of elastic bodies.
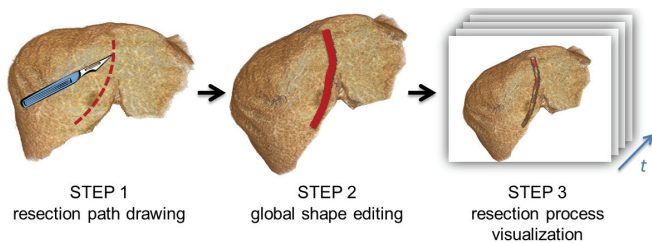
Figure 2 Direct volume manipulation framework for visualizing intraoperative liver resection process.

In this study, the local-frame-based elastic deformation model was developed for quickly computing the large deformation of elastic bodies resulting from twisting and bending. Our motivation is close to a CAD or a modeling framework that allows interactive editing of the volumetric object. From this point of view, stable, single-pass deformation that allows undo or backward operation is an essential factor. Therefore, the objective was to build a stable calculation model based on static analysis, while at the same time achieving local volume preservation.

3 DIRECT VOLUME MANIPULATION FRAMEWORK

In liver resection surgery, surgeons carry out ablation of liver tissue while checking reference vascular structures partially revealed in the process of resection. Important factors for planning and navigation of the liver resection surgery are the intraoperative deformed state of the liver and partial appearance of vessel structures. Visualization of entire vascular structures has been routinely used to grasp the global anatomy of a patient's liver. However, it does not provide an intraoperative view, that is, occluded anatomy and therefore has difficulty in comparing intraoperative views with visualization results. In addition to this, a practical design for clinical use is required for surgeons to project their image of surgical plans to the rendered volume directly and freely.

Considering these contexts and requirements, we designed a volume manipulation framework in which end users can directly deform volumetric objects and visualize the time-series resection process assumed in the surgery. Figure 2 shows a brief sketch of our manipulation framework.

STEP1 Drawing resection path

The user first defines a realistic resection path on the rendered image. We provide a mouse-based line drawing interface to support the input of the resection path. When the user indicates a pixel on the rendered image, our framework estimates the corresponding voxel in the volumetric space by accumulating opacity (or alpha) values at sample points in the eye direction. This process is similar to the ray casting protocol [35] commonly used in the volume rendering scheme. Once the accumulated opacity values exceed a threshold, we simply assume that the voxel at the current sample point has been selected by the user. This direct pointing scheme in a volumetric space was also employed in the volume sculpting study presented in [36].

A set of the sampled points $\mathbf{r}_i$ obtained from a drag operation represents a curved line (Figure 3a). By removing the surrounding voxels along the line, the user can input the next line to the internal structure of the liver. The curved surface S formed by the several input lines defines the global resection path (Figure 3b). Coexisting with the ray-casting-based pointing, we utilize MPR (multi planar reconstruction) to visualize cross sections of the liver. Inputting points $\mathbf{r}_i$ or a line on the MPR plane allows definition of a curved surface S while observing the internal structure of the liver. An example of a resection path is shown in Figure 3c and 3d.

STEP2 Global shape editing

The user edits the global shape of the rendered volume and visualizes his/her image of the intraoperative state of the liver. In order to achieve valid shape editing, we present an elastic model that stably represents large deformation due to rotation. The edited results are rendered with the resection path affected by global deformation. The details of the model are described in Section 4.

STEP3 Resection process visualization

The user finally visualizes the time-series resection process and reference vessels that are assumed to be partly observed in the surgery. Local deformation of the opening space generated by splitting liver tissue is represented to visualize internal structures and cross sections. The algorithms and implementation on GPU is explained in Section 5.

In our framework, we assume volume data $\mathbf{I}$ is a 3D scalar field composed of discrete voxels with an intensity value in a three-dimensional space. We apply a proxy geometry representation [9][37] and its specific data structure, the volume proxy mesh, for representing global shape editing. This mesh encodes volumetric geometry and physical information (i.e., elasticity and boundary conditions) between sampled vertices. Rather than being visualized directly, the mesh is placed into the entire volumetric area as a background data structure, thereby enabling a direct, interactive manipulation of the volume with elastic deformation.

The volume proxy mesh is formed with a standard tetrahedral mesh format $(\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ is sampled vertices and $\mathbf{E}$ is a set of tetrahedral elements. Three-dimensional manipulation and deformation of the target volume are supported by vertices placed in the volumetric space. In our framework, all model transformations with deformation are discretely modeled by displacement of vertices of the volume proxy mesh. First, all of
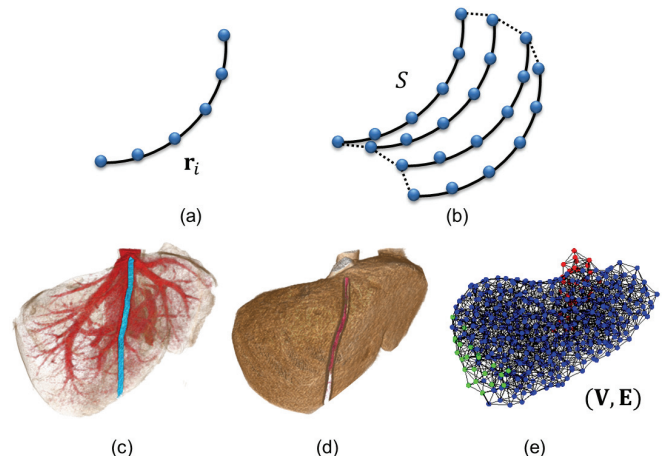

Figure 3 A resection path inputted on the rendered image. (a)(b) The global path is discretely represented as a set of points. (c)(d) Volumetric representation and (e) a volume proxy mesh with boundary condition as background data structure for shape editing
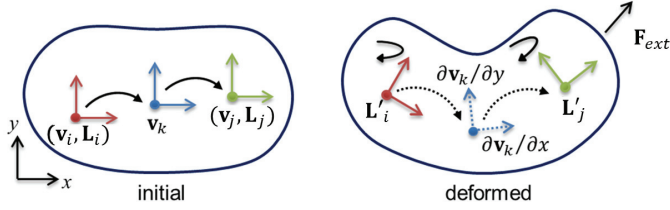
Figure 4 Local frame based deformation model. Linear elasticity defined on each local coordinate is evaluated to represent global deformation.

the vertices are categorized into three groups based on the physical conditions: *fixed* vertices (red), *constrained* vertices (green) and *free* vertices (blue) (see Figure 3e). The fixed vertices represent physically fixed structures that will not deform or move through manipulation, such as bones. The constrained vertices represent selected areas that can be directly controlled by the user. Lastly, the free vertices express other soft areas that will deform through manipulation. When the user manipulates the volume, initial vertices $\mathbf{V}$ are displaced to $\mathbf{V}'$ through mesh deformation. In the computation of deformation, the positions of fixed vertices and constrained vertices are imposed as physical constraints (or boundary conditions) that must be satisfied by the deformation model. Concurrently, free vertices are updated at every simulation step according to the computation results. The constrained vertices and free vertices represent model deformations. Next, our framework renders any deformed volume using the new coordinates of $\mathbf{V}'$ incorporating the initial volume intensity $\mathbf{I}$. If no manipulation occurs, $\mathbf{V}'$ is equivalent to the initial set of vertices $\mathbf{V}$. Note, again, that none of the vertices of the mesh are displayed to the user. This is the key point for enabling direct volume manipulation in our framework.

By integrating the mouse-based drawing interface [36] with the region growing process, the user can extract the volume of interest interactively. The related volume editing and segmentation techniques can be referred to in [8]. 2D drawing on the volumetrically rendered image is also used for setting constraints in shape editing. When the user draws a closed curve on the rendered volume, our framework sets the boundary condition to the selected vertices. Then, the user can manipulate the volume in 3D via 2D displacement of the constrained vertices. In our interface, the 2D displacement is first obtained by dragging the mouse and then transformed into 3D displacement parallel to the view-aligned slicing plane. This displacement information is used to set physical constraints, the external force given for the constrained vertices of the shape-editing model. By utilizing rotation of the scene, the user can interactively manipulate the volume in 3D. The mouse wheel is also used to enter the amount of twisting. The details of the force input are described in Section 4.

4 Shape Editing with Large Deformation

4.1 Elastic deformation based on local frames

This section introduces the proposed elastic model based on local frame theory for global shape editing of the liver (Figure 4). The goals for the model proposed in this study are as follows:

- To visualize plausible large elastic deformation
- to support a variety of manipulations such as twisting and bending in addition to pushing and pulling
- to preserve model volume
- to achieve fast, stable computation

Here, we do not aim to simulate the physically accurate behavior of real elastic objects but to support model deformation that satisfies the user's editing manipulation of the rendered image. Our motivation is close to a CAD or a modeling framework that allows interactive editing of the volumetric object. From this point of view, stable, single-pass deformation that allows undo or backward operation is an essential factor. Volume preservation is also important to satisfy the user's physical images of internal organs. Therefore, we employ a static analysis that does not consider the dynamic behavior of the elastic body. Consequently, we assume the external forces and the internal stresses produced by deformation are in equilibrium at every time step.

Figure 5 illustrates the algorithm flow for local-frame-based shape editing using the simplest line model as an example. The model has three vertices with different boundary conditions (see Figure 3e on the color setting). The 3DOF (degree-of-freedom) force $\mathbf{F}_{push}$ is used as an input to support pushing and pulling manipulation. In addition to this, the force applied when a small area of the object is twisted or rotated is modeled as $\mathbf{F}_{twist}$ in the rolling direction. (Figure 5a). A part of this pushing force $\mathbf{F}_{push}$ is used for computation of small deformations based on linear FEM. The other rotational factor is used to modify the local coordinates based on the moving frame theory in differential geometry to represent large deformation of the elastic body. When $\mathbf{F}_{push}$ and $\mathbf{F}_{twist}$ are given to the constrained vertex, the following four steps are processed to simulate the large deformation of elastic bodies.

STEP1 Force decomposition

As noted earlier, the linear finite element model is suitable for use in calculating deformation within the scope of small deformations. However, when the object is bent, $\mathbf{F}_{push}$ also affects the rotation of the object. This makes it impossible to correctly calculate the results of deformation [26]. With the presented method, on the other hand, the external force $\mathbf{F}_{push}$ in
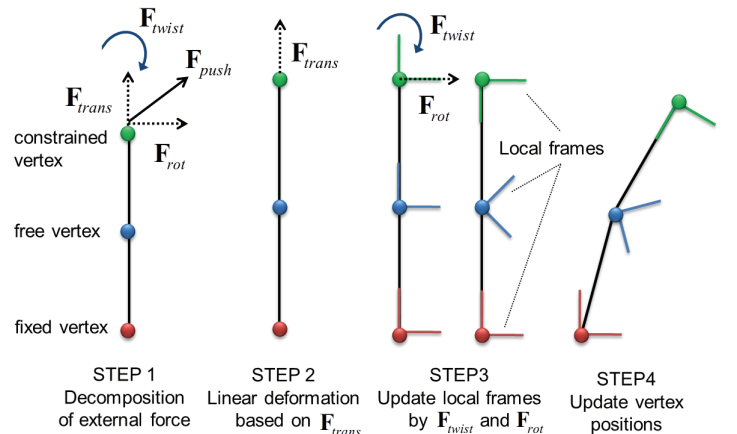

Figure 5 Local frame based deformation algorithms illustrated with a simple line model. The vertex position is determined by linear FEM, and rotation and scaling of the local frames.

the pushing/pulling direction is divided into two components: the component that will mainly contribute to small deformation of the object $\mathbf{F}_{trans}$, and the component contributing to the rotation of the object $\mathbf{F}_{rot}$. Currently, this decomposition is done using a skeleton vector determined by the fixed and constrained vertices.

STEP 2 Linear finite element deformation

The elastic deformation is evaluated using the translational component $\mathbf{F}_{trans}$ ($=\mathbf{F}_{push}-\mathbf{F}_{rot}$). The conventional linear FEM is used to compute the displacement of the free vertices under the boundary condition that $\mathbf{F}_{trans}$ is applied to the constrained vertices and positional constraints on the fixed vertices.

STEP 3 Update local frames

The rotational component $\mathbf{F}_{rot}$ and the initially applied $\mathbf{F}_{twist}$ in the rolling direction are converted into rotational constraints $\mathbf{R}_{rot}$ that are applied to the constrained vertices. All local frames defined at each vertex are rotated so as to satisfy these constraints. The small elastic deformation determined in STEP 2 is evaluated, and the local frames are corrected so as to preserve the volume locally.

STEP 4 Update vertex positions

Lastly, the vertex positions at the local coordinates that have been obtained are converted back to global coordinates to determine the final vertex positions.

Each step is processed by calculating static, linear equations, making it possible to determine a single deformation state for the given external force. This allows deformation to be simulated stably and interactively. Also, the undo operation is possible for each input force that is mapped to the corresponding deformation state. In the next section, the details of the computation for step 3 and 4 are explained.

4.2 Update of local frames and vertex positions

This section introduces the methods for updating local frames using $\mathbf{F}_{twist}$ in the rolling direction and the rotational component $\mathbf{F}_{rot}$ of the pushing force. In the local-frame-based mesh representation, a local frame is placed for each vertex. Instead of using the global coordinates, the three-dimensional position of the vertex is represented as a relative position at the local coordinates. The local frames placed at every vertex are also handled by the relative relationship with that of adjacent vertices.

The local frame at a vertex is handled as a 3×3 matrix and initialized to a unit matrix. When $\mathbf{F}_{twist}$ and $\mathbf{F}_{rot}$ are provided, the local frames and position of each vertex in the mesh are updated using the following three steps :

A. Rotate the local frames

In Step 1, the local frames placed at each vertex are updated. The rotation matrix $\mathbf{R}_{rot}\in R^{3\times3}$ is constructed using the $\delta|\mathbf{F}_{rot}|$ (δ : constant parameter) as a rotation angle. In the same manner, the rotation matrix $\mathbf{R}_{twist}\in R^{3\times3}$ is defined from the external force $\mathbf{F}_{twist}$. The local frames placed at all vertices are then rotated, with the two rotational amounts seen as a rotational constraints applied at the constrained vertices, and with the initial unit matrix seen as a fixed constraint applied to the fixed vertices. In this study, these constraints and the evaluation functions pertaining to the local coordinates are defined. The

optimal state is determined using the least squares approach. Equation (1) is the evaluation function to be solved.

$$\arg\min_{\mathbf{L}_i'}\sum_{j\in\mathbf{V}}\sum_{j\in N_i}\left\|(\mathbf{L}_j'-\mathbf{L}_i')-(\mathbf{L}_j-\mathbf{L}_i)\right\|^2+\sum_{k\in fixed}\varepsilon_k\left\|(\mathbf{L}_j'-\mathbf{L}_{fixed})\right\|^2 \quad(1)$$

Here, $\mathbf{L}_i$ is a 3×3 matrix that preserves base vectors as information of the local frames at $\mathbf{v}_i$. $\mathbf{L}_i'$ is the target matrix to be computed. The left term of the equation means the difference between the initial local frames and the target local frames. The Frobenius norm is used to calculate the norm of the matrix. The right term represents the difference between the constraints provided by the user and the initial local frames. The rotational constraints $\mathbf{R}_{rot}$ and $\mathbf{R}_{twist}$ given to the fixed and constrained vertices are applied to compute $\mathbf{L}_{fixed}$. N_i is the number of adjacent vertices at each vertex and ε_k is the weight. All of the local frames are obtained by computing $\mathbf{L}_i'$ which minimizes the evaluation function.

B. Scale the local frames

Next, if it is assumed that the volume of the elastic body is preserved entirely and locally as a result of deformation, the volume of each tetrahedron on the mesh should be preserved. In our framework, $\mathbf{F}_{trans}$ ($=\mathbf{F}_{push}-\mathbf{F}_{rot}$) is used as a boundary condition for reducing the artifact resulting from use of the linear finite element method. However, if a large pushing/pulling force is applied from the user, the artifact may be increased, producing changes in the volume. Accordingly, when the volume of the model has increased or decreased too much, the volume is corrected by scaling the axis of the local frames.

The Voronoi volume $\mathbf{V}_i$ defined for each vertex is used to evaluate the local volume of the elastic body. When the Voronoi volume $\mathbf{V}_i$ is changed to $\mathbf{V}_i'$ after computing STEP 2, the local frame axis is scaled using $(\mathbf{V}_i/\mathbf{V}_i')^{1/3}$ in order to achieve local volume preservation.

C. Update the vertex positions

Finally, the vertex positions are updated using the local frames that are updated through rotation and scaling. The vertex positions are calculated so as to minimize the displacement of the relative positions in the local coordinates before and after the update. In the same manner as Equation (1), the evaluation function is defined as follows.

$$\arg\min_{\mathbf{v}_i'}\sum_{j\in\mathbf{V}}\sum_{j\in N_i}\omega_i\left\|\mathbf{M}_i'(\mathbf{v}_j'-\mathbf{v}_i')-\mathbf{M}_i(\mathbf{v}_j-\mathbf{v}_i)\right\|^2+\sum_{k\in fixed}\left\|(\mathbf{v}_k-\mathbf{v}_{fixed})\right\|^2 \quad(2)$$

Here $\mathbf{v}_i$ is the initial vertex position and $\mathbf{v}_i'$ is the target vertex position after the deformation. $\mathbf{M}_i$ is the inverse matrix of the 3×3 matrix that represents the local frames at the vertex i. $\mathbf{v}_{fixed}$ is the positional constraint for the fixed vertex k. $\mathbf{L}_i'$ and $\mathbf{v}_i'$ are computed so as to minimize the evaluation functions defined by Equation (1) and (2) respectively. Since this is a linear least square problem, the coordinates $\mathbf{v}_i'$ for all vertices in the model after deformation can be determined by solving the linear equation.

5 RESECTION PROCESS VISUALIZATION

5.1 Local deformation based on resection path

This section describes techniques for locally visualizing vascular structures simulating intraoperative views. The reference vessels determined on preoperative CT volume become anatomical landmarks and key features which surgeons utilize to approach occluded tumors during surgery. Our framework simulates a time-varying space after local deformation of the partly resected liver. In order to represent discontinuous deformation around the resection path, we apply the concept of a volumetric inverse displacement map introduced by Correa et al [17] to our clinical application. However, as they use a predefined displacement map and do not describe details on the generation methods, we newly present a set of algorithms to generate a displacement map directly from the given resection path.

Our methods generate time-series displacement maps by computing per-voxel displacement vectors based on discrete points of the resection path (see Figure 2) and the voxel coordinate. A displacement map D is a forward map from voxel $\mathbf{p}$ to voxel $\mathbf{p'}$, that is, $\mathbf{p'} = \mathbf{p} + D(\mathbf{p})$ to visualize volume deformation. We store this map as a volumetric vector field, that is, three-dimensional vector array data with the same size in space and voxel resolution of the initial intensity volume. Figure 6 shows a 2D illustration for generating the displacement map and the outline of the algorithms is as follows:

1) The discrete points $\mathbf{R} = \{\mathbf{r}_0,...,\mathbf{r}_n\}$ are defined from the given global resection path which forms a curved surface $\mathbf{S}$. The normal vector $\mathbf{n}_i$ at the point $\mathbf{r}_i$ is computed. (Figure 6a)
2) For all voxels within a specified distance from $\mathbf{S}$, the distance between the voxel position $\mathbf{p}$ and the set of points $\mathbf{R}$ is searched. The closest point $\mathbf{r}_i$ from the voxel $\mathbf{p}$ is then determined (Figure 6b). A three-dimensional displacement vector $D(\mathbf{p})$ is computed based on a local displacement model described in the subsection 5.2. This computation is processed on GPU in parallel. (Figure 6c)
3) The inverse displacement map D^{-1}, a backward map from voxel $\mathbf{p'}$ to voxel $\mathbf{p}$, is generated from the displacement map D. This map is also quickly computed and interpolated on GPU to reduce visual artifacts, and used for referencing 3D texture in the final rendering process.

5.2 Local displacement model

Next we describe the local displacement model that defines the per-voxel displacement vector $D(\mathbf{p})$ based on the given curved surface $\mathbf{S}$. In order to shape an opening space between cross sections split by the ablation process, the displacement is determined by the distance from $\mathbf{S}$ and the probability density function on normal distribution. The displacement vector $D(\mathbf{p})$ at voxel $\mathbf{p}$ is defined as Equation (3).

$$D(\mathbf{p}) = \begin{cases} 0 & (vessel) \\ \alpha \exp\left(\dfrac{-(i-\mu)^2}{2\sigma^2} - \beta\mathbf{d}_{width} - \gamma\mathbf{d}_{depth}\right) & (others) \end{cases}$$

(3)

$$\mathbf{d}_{width} = (\mathbf{p} - \mathbf{r}_i) \cdot \mathbf{n}_i$$

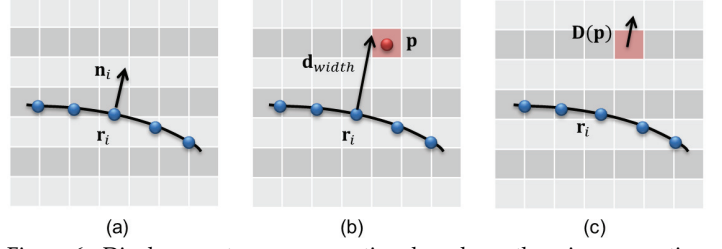$$\mathbf{d}_{depth} = (\mathbf{p} - \mathbf{r}_i) \cdot \mathbf{e}$$

Figure 6 Displacement map generation based on the given resection path.
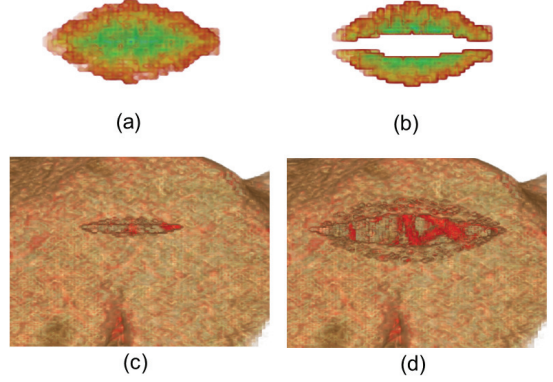

Figure 7 Examples of local deformation along resection path (a) displacement map, (b) inverse displacement map, (c) and (d) vascular structures visualized in the space after local deformation.

where μ and σ^2 is the average and the variance of the normal distribution, respectively. i is the index of the closest point of the curved surface $\mathbf{S}$. α is a parameter for controlling the width of the opening space. $\mathbf{d}_{width}$ approximates the vector from the closest point of $\mathbf{S}$ to the voxel $\mathbf{p}$ as shown in Figure 6b, and $\beta\mathbf{d}_{width}$ (β: a weight parameter) is used to reduce the displacement based on the distance from $\mathbf{S}$. $\gamma\mathbf{d}_{depth}$ plays a similar role to that of $\beta\mathbf{d}_{width}$ in the depth direction. $\mathbf{e}$ is the eye vector. A zero vector is set to $D(\mathbf{p})$ if the voxel $\mathbf{p}$ belongs to the vessel region, and the voxels for liver tissues are displaced. This makes it possible to visualize vascular structures in the opening space. As the vessel region generally has high intensity on CT data, a simple threshold can distinguish vascular structures from the liver volume.

The inverse displacement vector $D^{-1}(\mathbf{p'})$ is then computed from the displacement map. Using the inverse displacement map D^{-1}, $\mathbf{p}$ can be referred to as $\mathbf{p} = \mathbf{p'} + D^{-1}(\mathbf{p'})$. The 3D texture at the position $\mathbf{p}$ retrieves the color and opacity components. Finally, in order to handle discontinuities, it samples the alpha mask at the position $\mathbf{p'}$ and modulates the pixel's color components with the mask. In order to avoid aliasing artifacts in the cut, the program considers a point as transparent if the alpha mask is less than 0.5, and the resulting pixel will not contribute to compositing. Figure 7 shows colored examples of an inverse displacement map and local deformation results along a given resection path.

6 EVALUATION AND RESULTS

We implemented the proposed volume manipulation framework using C++, GLSL and CUDA. We used the tetgen library (published at http://tetgen.berlios.de/) for creating the tetrahedral elements of the mesh based on CT data. The recent
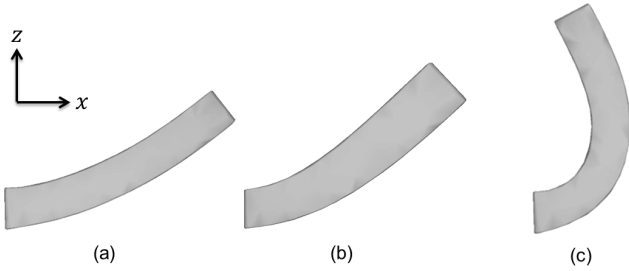
(a)   (b)   (c)

Figure 8   Deformation results of an elastic bar model. (a)(c) proposed shape editing and (b) linear FE-based deformation with volume error.
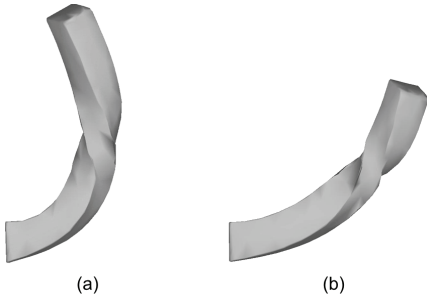

(a)   (b)

Figure 9   Large deformation results with twisted rotation. (a) Homogeneous configuration (1.0MPa) and (b) heterogeneous configuration (left: 2.0MPa and right: 1.0MPa)
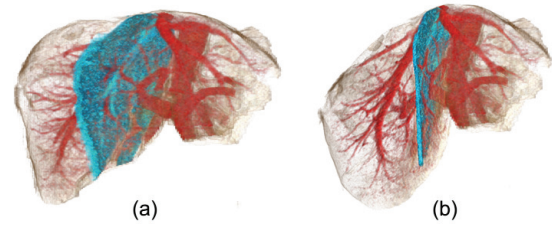

(a)   (b)

Figure 10 (a) Visualization of global resection path and (b) a shape editing result manipulated by a surgeon. The assumed intraoperative state of the liver is visualized with internal vascular structures.

Figure 11 Shape editing examples of the liver volume. The proposed local-frame-based model can produce visually valid editing results. This image rouses intraoperative resection path that should be traced.

meshing techniques or software can be referred to in [38] and in [39]. The quality and performance were examined using both public domain data and clinical CT data. We used a standard PC (CPU: Intel Corei7 2.93 GHz, Memory: 16GB) with a general-purpose graphic card (nVidia GeForce GTX580) for the test.

6.1   Shape editing with large deformation

The results of deformation using the proposed elastic model were first validated as a quantitative indicator of the change in the volume of a model shape in response to the application of external force. The visual naturalness of the results of large deformation was also confirmed qualitatively. A prepared bar model shown in Figure 8 has 507 vertices and 1750 tetrahedral elements. 1.0 MPa was uniformly applied to the elastic parameter of each element. The results of deformation based on a conventional linear finite element (FE) model were used for comparison.

We obtained some results of deformation when an external force is applied in several stages diagonally upward (z-direction) to the right (x-direction) at one end of the bar model. In the case of deformation using a conventional FE-based model, the model expands as the external force increases, leading to deformation results that appear unnatural. The volume increases to 144% and 274% when an external force of 150N and 300N was applied in z-direction respectively. In contrast, with the proposed model, deformation can be represented naturally even when a great external force is applied. Changes in volume can also be kept to 2 - 3% or less (see Table 1).

Table 1   Volumes of the deformed elastic bar model.

		force x		
		0 N	300 N	500 N	700 N
force z	0 N	100.0%	100.0%	100.1%	100.1%
	100 N	99.1%	99.2%	99.2%	98.2%
	200 N	98.1%	98.1%	98.2%	98.2%
	300 N	97.5%	97.6%	97.6%	97.6%

Another test was conducted to determine whether or not physical property is reflected in bending and twisting deformation. Deformation of the bar model was conducted for two cases (one in which the Young's modulus of the model was set uniformly at 1.0 MPa for the entire model, and another in which the Young's modulus was set to 2.0 MPa for the left half area and 1.0 for the right half area) and the results of deformation were compared. Figure 9 shows the results of the simulation. In the case of the heterogeneous model shown in (b), larger deformation was produced in the right side of the model, and heterogeneous deformation that reflects differences in the Young's modulus in places can be confirmed visually.

6.2   Resection process visualization

The next experiment was done with a surgeon in department of hepato-biliary-pancreatic and transplant surgery. Using CT data and assuming actual surgery, a resection path was defined on the segmented liver volume. (Figure 10a) A volume proxy mesh with 317 vertices and 1034 tetrahedral elements was constructed from the liver volume. 1.0 MPa was set for Young's modulus and 0.4 was set for Poisson's ratio. The boundary condition for shape editing was set to vertices as shown in Figure 3e. Then, the surgeon manipulated the liver shape based on his image of the surgery. Figure 10b shows a shape editing result with the given resection path on the segmented liver volume. He checked the deformation of internal vascular structures and finally fixed the assumed intraoperative state of the liver. Figure 11 shows his manipulation from the initial state. In

tis manipulation, rotational operation was carried out by setting the rotation axis near the aorta. The left one is the path in the non-deformed liver, and the other two are the same paths on the liver deformed with different rotation angles: 105° and 135°. These figures demonstrate how the resection path is affected by global deformation of the liver.

Lastly, the resection process was simulated based on the planned path (see Figure 12). Local deformation along the resection path could visualize a part of internal vascular structures using inverse displacement mapping. The sequential resection process was visualized by changing the depth parameter of the given resection path. As shown in Figure 12, a thin vessel first appeared. Increasing the depth revealed a middle hepatic vein and then a portal vein appeared. These vascular structures are utilized as references which is key information when approaching occluded tumors during surgery. Thus, time-series visualization of the local appearance of vascular structures based on the real surgical process could be significant information to support surgical ablation.

6.3　Performance

The performance of the proposed methods was measured. As we employed texture-based volume rendering for visualizing proxy geometries [9], volume data are handled as a 3D texture. In this approach, tetrahedral elements are sampled to render the deformed results with the 3D texture. The number of vertices in the proxy geometry affects the computation time more than the volume size does. Therefore, we prepared three tetrahedral meshes with different vertices, and measured the computation time. The number of vertices was chosen after considering the previous experiment, the variety of the liver's shape and the specialist's opinion on the deformation quality. In this test, the CT liver data with 256^3 voxels was used. The average computation time for deformation and volume rendering is shown in Table 2. In these cases, over 10 FPS (frames per second) was obtained, and it was possible to conduct natural, interactive operations. The presented deformation model computes the linear equations twice. Since our approach does not require integral calculations, and deformation results can be determined at each step, stable shape editing is achieved even for large deformations.

In order to generate sequential images shown in Figure 12, $D(\mathbf{p})$ and $D^{-1}(\mathbf{p}')$ are computed on GPU based on the given resection path. This process is a per-voxel operation and currently requires about 1.4 seconds even if parallel computing is used. Therefore, we pre-computationally generate the required sequential images by changing the weight parameters. Once $D^{-1}(\mathbf{p}')$ is prepared, the user can interactively deform the volume with the resection path. The resection process in the deformed state can be previewed.

Vertices	Deformation	Rendering	Total
317	28.7	24.6	53.4
424	39.0	38.4	77.4
507	50.8	47.6	98.5

Table 2　Average computation time (msec) for deformation and rendering
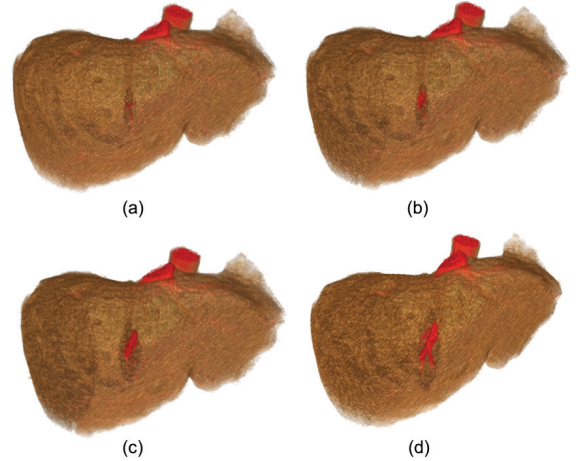

Figure 12　Sequential visualization of the resection process by local deformation along the resection path. (a) thin vessels, (b)(c) a middle hepatic vein and (d) a portal vein appeared in the opening space.

6.4　Evaluation and Discussion

In the tests, we asked the specialists on the usability and limitation of the developed system. Specialists used the developed system commented that the model could:

- Support intuitive preoperative planning of patient-specific resection path shapes on the rendered image
- Express a part of vascular structures and similar local views during the ablation process
- Be applied when sharing a "roadmap" of vascular structures that should be traced with other medical staff
- Be useful for intraoperative navigation

The specialists also suggested that quantitative evaluation would support the effectiveness of the intraoperative navigation. As this work aimed to visualize the intraoperative state of the liver based on the surgeon's image, anatomical knowledge and experience, we rather focus on providing stable, real-time computation of the shape editing for efficient preoperative planning. Therefore, we note physical accuracy of the deformation model should be further verified for use in surgical navigation. The deformation results may be strengthened if image-based or tracking-based registration with estimated/measured elastic parameters [40][41] is integrated into our framework.

We have also found some limitations from the current volume manipulation framework. For example, extremely large deformation can yield visual artifacts due to deformation of the volume being represented by sparsely sampled tetrahedral meshes because the inside of the tetrahedral elements are linearly interpolated. In the same fashion, heterogeneous elasticity modeling is currently restricted to very simple objects because the volume is sparsely sampled using hundreds of vertices. The displacement map is used to represent the ablation process. Although this approach is useful for visualizing partial cuts, simulating cutting through an organ with large deformation require additional modeling such as tetrahedral subdivision. This is basically the trade-off problem between complex topology modeling and interactive performance of overall algorithms. These issues will be our future work.

7 CONCLUSION

This paper proposed a new design and application of direct volume manipulation to visualize the intraoperative liver resection process. Handling interactive volume deformation and resection simultaneously has been a technical hurdle. To address this issue, our framework modeled global shape editing and discontinuous local deformation by merging proxy geometry encoding and displacement mapping. The local-frame-based elastic model was presented to allow stable editing of the liver shape during rotation while preserving the volume. Several tests confirmed the developed software and interface visualize the intraoperative state of liver and produce local views of reference vascular structures, which provides a "road map of vessels" that are key features when approaching occluded tumours.

We hope the presented techniques are applied to more complex volume data and theoretically contribute to volume visualization research. We continue to improve the framework and to develop practical medical applications for preoperative planning and for intraoperative navigation. Specifically, enhancing reality of both deformable models and real-time rendering are our future work.

ACKNOWLEDGEMENTS

REFERENCES

[1] E. K. Fishman, B. S. Kuszyk, D. G. Heath, G. Luomin and B. Cabral, Surgical planning for liver resection, IEEE Computer, Vol. 29, No. 1, pp. 64-72, 1996.

[2] B. Reitinger, A. Bornik, R. Beichel and D. Schmalstieg, Liver surgery planning using virtual reality, IEEE Computer Graphics and Applications, Vol. 26, No. 6, pp. 36-47, 2006.

[3] M. Scheuering, A. Schenk, A. Schneider, B. Preim and G. Greiner, Intraoperative augmented reality for minimally invasive liver interventions, Proc. SPIE, Vol. 5029, 407, 2003.

[4] O. Heinzmann, S. Zidowitz, H. Bourquain, S. Potthast, H.O. Peitgen, D. Oertli and C. Kettelhack, Assessment of intraoperative liver deformation during hepatic resection: Prospective Clinical Study, World J Surg, Vol. 34, No. 8, pp. 1887-1893, 2010.

[5] D. M. Cash, M. I. Miga, S. C. Glasgow et al, Concepts and preliminary data toward the realization of image-guided liver surgery, J Gastrointest Surg, Vol. 11, pp. 844–859, 2007.

[6] C. Hansen, S. Zidowitz, A. Köhn, M. Hindennach and H. Peitgen, Intraoperative adaptation and visualization of preoperative risk analyses for oncologic liver surgery, Proc. SPIE, Vol. 6918, 691809, 2008.

[7] S. Bruckner and M. E. Groller, Exploded views for volume data, IEEE Trans. on Visualization and Computer Graphics, Vol. 12, No. 5, pp. 1077-1084, 2006.

[8] K. Burger, J. Kruger and R. Westermann, Direct volume editing, IEEE Trans. on Visualization and Computer Graphics, Vol. 14, No. 6, pp.1388-1395, 2008.

[9] M. Nakao, K. W. C. Hung, S. Yano, K. Yoshimura and K. Minato, "Adaptive Proxy Geometry for Direct Volume Manipulation", IEEE Pacific Visualization, pp.161-178, 2010

[10] C. D. Correa, D. Silver and M. Chen, Feature aligned volume manipulation for volume illustration, IEEE Transactions on Visualization and Computer Graphics, Vol. 12, No. 5, pp.1069-1076, 2006.

[11] C. D. Correa, D. Silver and M. Chen, Volume deformation via scattered data interpolation, Proc. of the 6th Eurographics / IEEE VGTC Workshop on Volume Graphics, pp. 91-98, 2007.

[12] V. Singh, D. Silver and N. Cornea, Real-time volume manipulation, Proc. of Eurographics/IEEE TVCG Workshop on Volume graphics, pp. 45-51, 2003.

[13] S. Fang, S. Huang, R. Srinivasan and R. Raghavan, Deformable volume rendering by 3D texture mapping and octree encoding, Proc.of IEEE Visualization, pp. 73-80, 1996.

[14] W. Lin and R. A. Robb, Dynamic volume texture mapping and model deformation for visually realistic surgical simulation, Proc. Medicine Meets Virtual Reality, pp. 198-204, 1998.

[15] M. J. Mcguffin, L. Tancau and R. Balakrishnan, Using deformations for browsing volumetric data, Proc. of IEEE Visualization, pp.401-408, 2003.

[16] M. Chen, D. Silver, A. Winter, V. Singh and N. Cornea, Spatial transfer functions: A unified approach to specifying deformation in volume modeling and animation, Proc.of Volume Graphics, pp. 35-44, 2003.

[17] C. D. Correa, D. Silver and M. Chen, Discontinuous displacement mapping for volume graphics, Eurographics / IEEE VGTC Workshop on Volume Graphics, pp. 9-16, 2006.

[18] C. Rezk-Salama, M. Scheuering, G. Soza and G. Greiner, Fast volumetric deformation on general purpose hardware, Proc. of The ACM SIGGRAPH / EUROGRAPHICS workshop on Graphics hardware, pp.17-24, 2001.

[19] R. Westermann and C. Rezk-Salama. Real-time volume deformations, Computer Graphics Forum, Vol. 20, No. 3, pp.443-451, 2001.

[20] Y. Masutani, Y. Inoue, K. Ishii, N. Kumai, F. Kimura and I. Sakuma, "Development of surgical simulator based on fem and deformable volume rendering", Proc. SPIE, pp.500-507, 2004.

[21] J. Georgii and R. Westermann, A generic and scalable pipeline for GPU tetrahedral grid rendering, Proc. IEEE Visualization, pp.1345-1352, 2006.

[22] J. Berkley, G. Turkiyyah, D. Berg, M. Ganter and S. Weghorst, "Real-time finite element modeling for surgery simulation: an application to virtual suturing", IEEE Trans. on Visualization and Computer Graphics, Vol. 10, No. 3, pp.314-325, 2004.

[23] M. Bro-Nielsen and S. Cotin, "Real-time Volumetric de-formable Models for Surgery Simulation using Finite Elements and Condensation", Computer Graphics Forum, Vol. 15, No. 3, pp.57-66, 1996.

[24] U. Meier, O. Lopez, C. Monserrat, M.C. Juan, M. Alcaniz, "Real-time deformable models for surgery simulation: a survey", Computer Methods and Programs in Biomedicine, Vol. 77, No. 3, pp. 183-197, 2005.

[25] A. Hölzer, C. Schröder, M. Woiczinski, P. Sadoghi, A. Scharpf, B. Heimkes, V. Jansson, "Subject-specific finite element simulation of the human femur considering inhomogeneous material properties: a straightforward method and convergence study", Computer Methods and Programs in Biomedicine, Vol. 110, No. 1, pp. 82-88, 2012.

[26] A. Nealen, M. Muller, R. Keiser, E. Boxerman and M. Carlson, "Physically based deformable models in computer graphics", Proc. of Eurographics, pp. 809-836, 2005.

[27] J. Barbic and D. L. James. Real-time subspace integration for St. Venant-Kirchhoff deformable models, ACM Trans. Graphics, Vol. 24, No. 3, pp. 982–990, 2005.

[28] S. Niroomandi, I. Alfaro, E. Cueto, F. Chinesta, "Accounting for large deformations in real-time simulations of soft tissues based on reduced-order models", Computer Methods and Programs in Biomedicine, Vol. 105, No. 1, pp. 1-12, 2012.

[29] R. Kikuue, H. Tabuchi and M. Yamamoto, "An Edge-Based Computationally-Efficient Formulation of Saint Venant-Kirchhoff Tetrahedral Finite Elements", ACM Trans on Graphics, Vol. 28, Issue 1, 2008.

[30] M. Muller, J. Dorsey, L. McMillan, R. Jagnow, and B. Cutler, "Stable real-time deformations", Proc. of ACM SIG-GRAPH/Eurographics Symposium on Computer Animation, pp. 49-54, 2002.

[31] M. Muller and M. Gross, "Interactive Virtual Materials", Proc. Graphic Interface, pp. 239-246, 2004.

[32] M. Botsch and O. Sorkine, "On linear variational surface deformation methods", IEEE Trans. on Visualization and Computer Graphics, Vol. 14, No. 1, pp.213-230, 2008.

[33] Y. Lipman, D. Cohen-Or, R. Gal, D. Levin, "Volume and shape preservation via moving frame manipulation", ACM Trans. on Graphics (SIGGRAPH), Vol. 26, No. 1, 2007.

[34] Y. Lipman, O. Sorkine, D. Levin and D. Cohen-Or, "Linear rotation-invariant coordinates for meshes", Proc. ACM SIGGRAPH, pp. 479-487, 2005.

[35] M. Levoy, "Efficient ray-tracing of volume data", ACM Trans. on Graphics, Vol. 9, No. 3, pp.256-261, 1990.

[36] K. Imanishi, M. Nakao, M. Kioka, M. Mori, M. Yoshida, T. Takahashi and K. Minato, "Interactive Bone Drilling using a 2D Pointing Device to Support Microendoscopic Discectomy Planning", International Journal of Computer Assisted Radiology and Surgery, Vol. 5, No. 5, pp. 461-469, Sep. 2010.

[37] K. W. C. Hung, M. Nakao, K. Yoshimura and K. Minato, "Background-incorporated Volumetric Model for Patient-Specific Surgical Simulation: A Segmentation-free, Modeling-free Framework", Int. J. Computer Assisted Radiology and Surgery, Vol. 6, No. 1, pp. 35-45, 2011.

[38] Z . Yu, J. Wang, Z. Gao, M. Xu, M. Hoshijima, "New software developments for quality mesh generation and optimization from biomedical imaging data", Computer Methods and Programs in Biomedicine, Vol. 113, No. 1, pp. 226-240, 2014.

[39] M. Meyer, R. Whitaker, and M. Kirby, "Particle-based Sampling and Meshing of surfaces in Multimaterial Volumes", IEEE Transactions on visualization and computer graphics. Vol. 14, No.6, pp. 1539-1546, 2008.

[40] A. Hostettler, D. George, Y. Rémond, S. Nicolau, L. Soler, J. Marescaux, "Bulk modulus and volume variation measurement of the liver and the kidneys in vivo using abdominal kinetics during free breathing", Computer Methods and Programs in Biomedicine, Vol. 100, No. 2, pp. 149–157, 2010.

[41] F. Martínez-Martínez, M. J. Rupérez, J. D. Martín-Guerrero, C. Monserrat, M.A. Lago, E. Pareja, S. Brugger, R. López-Andújar, "Estimation of the elastic parameters of human liver biomechanical models by means of medical images and evolutionary computation", Computer Methods and Programs in Biomedicine, Vol. 111, No. 3, pp. 537-549, 2013.