Computer-Aided Design 42 (2010) 95–108

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Physically based modeling and simulation with dynamic spherical volumetric simplex splines

Yunhao Tan[a], Jing Hua[a,*], Hong Qin[b]

[a] *Department of Computer Science, Wayne State University, Detroit, MI 48202, USA*

[b] *Department of Computer Science, Stony Brook University, Stony Brook, NY 11794, USA*

ARTICLE INFO

Article history:
Received 26 June 2008
Accepted 20 February 2009

Keywords:
Physically based modeling
Heterogeneous models
Biomechanic simulation
Biomedical imaging

ABSTRACT

In this paper, we present a novel computational modeling and simulation framework based on dynamic spherical volumetric simplex splines. The framework can handle the modeling and simulation of genus-zero objects with real physical properties. In this framework, we first develop an accurate and efficient algorithm to reconstruct the high-fidelity digital model of a real-world object with spherical volumetric simplex splines which can represent with accuracy geometric, material, and other properties of the object simultaneously. With the tight coupling of Lagrangian mechanics, the dynamic volumetric simplex splines representing the object can accurately simulate its physical behavior because it can unify the geometric and material properties in the simulation. The visualization can be directly computed from the object's geometric or physical representation based on the dynamic spherical volumetric simplex splines during simulation without interpolation or resampling. We have applied the framework for biomechanic simulation of brain deformations, such as the brain shifting during surgery and brain injury under blunt impact. We have compared our simulation results with the ground truth obtained through intra-operative magnetic resonance imaging and real biomechanic experiments. The evaluations demonstrate the excellent performance of our new technique.

1. Introduction

Modeling, simulation and assessment of digital representations of heterogeneous objects acquired from the real world are very challenging research tasks and have many potential applications. The fundamental objectives are to unambiguously model high-dimensional heterogeneous objects, accurately and effectively simulate their behaviors, and rigorously analyze their dynamic natures. Among many important aspects of physically based modeling and simulation, accuracy is of the utmost importance since only physically realistic simulations can be used to represent true reality and provide valuable information for simulation-based assessment and analysis. In existing approaches, several different representations are typically required throughout the simulation of real-world models in computerized environments. That is to say, each stage within the entire physical simulation pipeline, including modeling (e.g., meshing, material modeling), simulation, analysis, visualization, typically takes as input a different representation of the modeled object, which requires costly and error-prone data conversions throughout the entire simulation process. It will certainly introduce error into the pipeline. For instance, in order to simulate the brain deformation, a linear solid mesh needs to be generated for finite element methods (FEMs) from the voxel-based representation of the brain representing the geometry of the brain (which has a highly convoluted cortical surface and many subtle sub-cortical structures). Then, manual material editing needs to be conducted to assign material properties to solid meshes. The FEM properties are linearly interpolated during simulation and resampled once again to the voxels' intensities for visualization. Certainly, conversions among volumetric datasets, solid meshes, finite elements, and voxels based on linear interpolation or resampling will introduce error. In addition, more errors will be brought into the pipeline as the constructed linear solid mesh may not well represent both geometry and material distribution simultaneously. The geometric, physical, and mechanical properties are not tightly integrated into the simulation. As a result, current practice impedes the accurate modeling and simulation of digital models of real-world objects. With ever-improving computing power comes the strong demand for more accurate, robust, and powerful solid modeling and simulation paradigms that are efficacious for the modeling, simulation, analysis, and visualization of digital models of real-world objects.

* Corresponding address: Department of Computer Science, Wayne State University, 5143 Cass Ave, 431 State Hall, Detroit, MI 48202, USA. Tel.: +1 313 577 9004; fax: +1 313 577 6868.

E-mail address: jinghua@cs.wayne.edu (J. Hua).

0010-4485/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2009.02.014

In order to bridge the gap and overcome the aforementioned deficiencies, we develop an integrated computational framework based on dynamic spherical volumetric simplex splines (DSVSS) that can greatly improve the accuracy and efficacy of modeling and simulation of heterogenous objects since the framework cannot only reconstruct with high accuracy geometric, material, and other quantities associated with heterogeneous real-world models, but also simulate the complicated dynamics precisely by tightly coupling these physical properties into the simulation. The integration of geometric modeling, material modeling, and simulation is the key to the success of the simulation of real-world objects. In contrast to existing techniques, our framework uses a single representation that requires no data conversion. The advantages of our framework result from the many attractive properties of multivariate splines. In comparison with tensor-product NURBS, multivariate simplex splines are non-tensor-product in nature. They are essentially piecewise polynomials of the lowest possible degree and the highest possible continuity everywhere across their entire tetrahedral domain. For example, given an object of simplex splines with degree n, it can achieve C^{n-1} continuity. Furthermore, C^0, other varying continuities, and even discontinuity can be accommodated through different knot and control point placements and/or different arrangements of domain tetrahedra in 3D. Furthermore, simplex splines are ideal to represent heterogeneous material distributions through the tight coupling of control points and their attributes. From a dynamic simulation's point of view, they are finite elements which can be directly brought into finite element formulations and physics-based analysis without losing any information. Finite elements can be derived directly from the simplex spline representation, which can also be visualized via volumetric ray-casting without discretization [1]. Trivariate simplex splines are obtained through the projection of n-dimensional simplices onto 3D. Projecting them one step further onto 2D for visualization results in bivariate simplex splines of one degree higher than the original solid model, therefore, simplex splines facilitate the visualization task with an analytical, closed-form formulation. It is not necessary to perform any resampling and/or interpolation operations. Local adaptivity and local/global subdivision via knot insertion can be readily achieved.

On the application front, in recent years, tremendous efforts from biomedical research communities have been devoted into brain simulation since accurate simulation of brain deformations can have many potential applications, e.g., computer-aided surgical planning/surgery, computer-assisted disease/injury positioning, accurate radiation therapy, and many other medical benefits [2]. Various methods are emerging for the simulation of brains in different physical environments. However, most brain volume simulation techniques still depend on linear geometric representation and FEMs as we have already described above. No advanced computational models are available for better simulation. As we all know, the brain is a highly convoluted organ rich with geometric, anatomical, and material variations. In order to obtain a realistic deformation simulation of the brain, it is very important to construct a digital model which can simultaneously represent its geometry, imaging intensities, and material properties, and then integrate the properties into the biomechanic simulation. Consider that the human brain is topologically equivalent to a solid sphere, our proposed dynamic spherical volumetric simplex splines are perfect for modeling, simulation, and analysis of such an object. The spherical volumetric simplex splines are defined over a solid spherical tetrahedralization. In this paper, we apply and evaluate our simulation framework on various human brain deformations.

As depicted in Fig. 1, the developed framework is fully automated without human intervention. The spherical domain is constructed from the subdivision of an icosahedron and harmonic volumetric mapping. With spherical domain and harmonic volume parameterization, the continuous volumetric representation of the modeled object is obtained through fitting spherical volumetric simplex splines to the real-world volume data. Physical properties can then be integrated into the system to unify the geometric representation as well as the physical representation. With Lagrangian dynamics essentials integrated into the pipeline, the powerful framework yields the dynamic representation of the digital model. The dynamic representation of the digital model can facilitate multiple tasks such as model assessment, biomechanic simulation, and visualization.

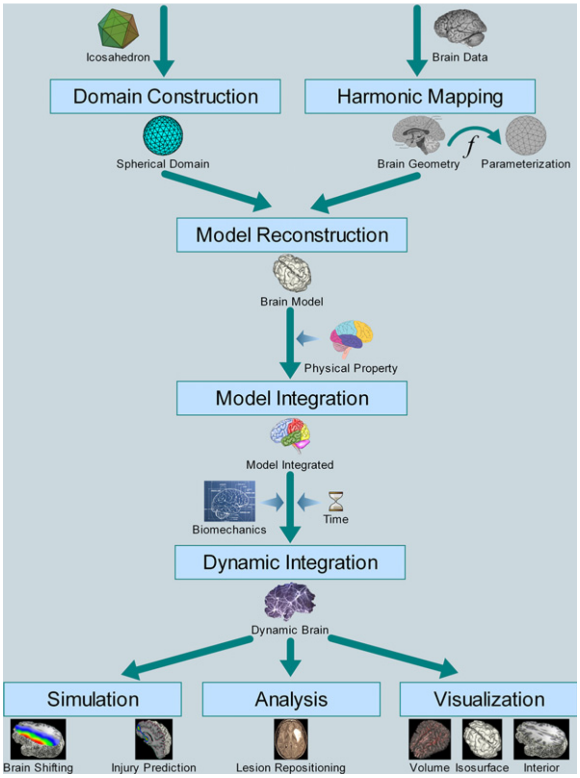

Fig. 1. Illustration of the pipeline of our DSVSS framework. The rectangles inside the pipeline indicate tasks, and the icons with text under them denote the inputs and outputs of tasks.

Our contributions in this paper can be summarized as follows:

- We develop a physical simulation framework which seamlessly integrates geometric properties, physical properties, and dynamic behaviors together. The consistent, uniform representation throughout each stage of modeling and simulation is a single degree n spherical volumetric simplex spline. It is ideal for simulating complex, heterogenous real-world objects.
- The heterogenous model reconstructed from the digitalization of a real-world object is faithful and of high-fidelity in terms of its geometry and material distribution. The model reconstruction procedure is automatic, and the maximal fitting error to the original data can be controlled by user's specification interactively.
- During the simulation, the geometry and physical properties of the volumetric model can be computed using the analytic representation without any need for numerical approximations such as cubic interpolation or quadratic resampling. Hence, physical simulation, including all downstream processes, such as analysis and evaluation, can be achieved more accurately and robustly.
- We apply the dynamic spherical simplex splines scheme in the simulation and analysis of brain models. The unified scheme can

achieve a very accurate simulation compared with the ground-truth results because it can tightly integrate the geometric and material properties in the simulation. Our framework has great potential to provide simulation-based assessment for innovative computer-aided diagnosis of brain injury cases.

2. Previous work

This section reviews the previous work related to the theory and application of multivariate simplex splines and physically based modeling and simulation. In particular, we provide a brief background regarding brain simulation and its potential applications.

2.1. Multivariate simplex splines

From a projection's point of view, univariate B-splines can be intuitively formulated as volumetric shadows of higher dimensional simplices, i.e., we can obtain B-splines of arbitrary degree n by taking a simplex in the $(n+1)$-dimensional space and volumetrically projecting it onto R^1. Motivated by this idea of Curry and Schoenberg, C. de Boor [3] presented a brief description of multivariate simplex splines. In essence, multivariate simplex splines are the volumetric projection of higher dimensional simplices onto a lower dimensional space R^m. Simplex splines have many attractive properties such as piecewise polynomials over general tetrahedral domains, local support, higher-order smoothness, and positivity, making them potentially ideal in engineering design applications [4]. From the point of view of blossoming, Dahmen et al. [5] proposed triangular B-splines. Later, Greiner and Seidel [4] demonstrated their practical feasibility in graphics and shape design.

In contrast to theoretical advances, the application of simplex splines has been rather under-explored. Pfeifle and Seidel developed a faster evaluation technique for quadratic bivariate DMS-spline surfaces [6] and applied it to the scattered data fitting of a triangular B-spline [7]. Recently, Rössl et al. [8] presented a novel approach to reconstruct volume from structure-gridded samples using trivariate quadric super splines defined on a uniform tetrahedral partition. They used Bernstein–Bézier techniques to compute and evaluate the trivariate spline and its gradient. Hua and Qin presented a volumetric sculpting framework that employs trivariate scalar nonuniform B-splines as an underlying representation [9,10]. More recently, they applied trivariate simplex splines to the representation of solid geometry, the modeling of heterogeneous material attributes, and the reconstruction of continuous volumetric splines from discretized volumetric inputs via data fitting [11]. Tan et al. applied the hierarchical simplex splines to volume reconstruction from planar images [12].

2.2. Physically based modeling and biomechanic simulation

Free-form deformable models were first introduced to the modeling community by Terzopoulos et al. [13], and they have been improved by a number of researchers over the past 20 years. Celniker and Gossard developed an interesting prototype system [14] for interactive free-form design based on the finite-element optimization of energy functionals proposed in [13]. Bloor and Wilson developed related models using similar energies and numerical optimization [15]. Welch and Witkin extended the approach to trimmed hierarchical B-splines for interactive modeling of free-form surfaces with constrained variational optimization [16]. Terzopoulos and Qin [17,18] devised dynamic physically based generalization of NURBS (D-NURBS). Later, they further developed a dynamic triangular B-splines [19] paradigm for high topology surface modeling. The new paradigm on simplex spline finite elements is substantially more sophisticated and is expected to produce even more true-to-life simulation results.

As for simulation of digital models of real-world objects, researchers have focused on FEM meshing, which can represent the shape of the objects, and physical laws and properties, which govern the model's behavior. Zhang et al. presented a method for 3D mesh generation from imaging data [20]. They further designed an algorithm for automatic 3D mesh generation for a domain with multiple materials. In general, the main objective of FEM meshing is to construct nicely-shaped elements which can represent both the geometry and material of real-world models for accurate and robust simulation. However, due to its linear representations in general, it cannot accurately represent the geometric and physical properties of real-world objects. For simulation-based assessment of real-world objects, e.g., the brain, these FEM representations are not able to obtain an accurate and objective analysis result [21].

Biomechanic simulation of brain behaviors such as brain shifting and brain injury gains ever-increasing importance in recent years while these behaviors remain an unclear problem for public health professionals. Although impeded by that fact that brain material properties cannot be retrieved directly from the human brain in vivo, there is a certain amount of research which has been done either using animal brains or modifying brain biomechanic parameters to approach the real situation. Margulies et al. studied the relationship between non-preconditioned and preconditioned biomechanic response of brain tissue from pigs [22]. Later they further investigated the homogeneity of gray matter by measuring the stiffness of the cerebral cortex and comparing it to the thalamus of the porcine brain [23]. Many investigations have been conducted using mathematical finite element modeling [24,25,21]. In general, the cerebral tissues in their models were represented by homogeneous materials. Recent studies started to make distinctions between gray and while matter. In terms of applications, brain deformation simulation facilitates researchers and clinicians new prospects in clinical practice [2].

3. Dynamic spherical volumetric simplex splines

In this section, we first briefly review the theoretical background of volumetric simplex splines. Then, we formalize them to the spherical volumetric simplex splines with details on spherical domain construction. We further generalize the splines with physical dynamics and develop dynamic spherical simplex splines which can be used for modeling and simulation of real-world models.

3.1. Volumetric simplex splines

A degree n volumetric simplex spline, $M(\mathbf{x}|\mathbf{x}_0,\ldots,\mathbf{x}_{n+3})$, can be defined as a function of $\mathbf{x}\in\mathbb{R}^3$ over the half open convex hull of a point set $\mathbf{V}=[\mathbf{x}_0,\ldots,\mathbf{x}_{n+3})$, depending on the $n+4$ knots $\mathbf{x}_i\in\mathbb{R}^3$, $i=0,\ldots,n+3$. The volumetric simplex splines may be formulated recursively, which facilitates point evaluation and its derivative and gradient computation. When $n=0$,

$$M(\mathbf{x}|\mathbf{x}_0,\ldots,\mathbf{x}_3)=\begin{cases}\dfrac{1}{|\mathrm{Vol}_{\mathbb{R}^3}(\mathbf{x}_0,\ldots,\mathbf{x}_3)|}, & \mathbf{x}\in[\mathbf{x}_0,\ldots,\mathbf{x}_3),\\ 0, & \text{otherwise},\end{cases}$$

and when $n>0$, select four points $\mathbf{W}=\{\mathbf{x}_{k_0},\mathbf{x}_{k_1},\mathbf{x}_{k_2},\mathbf{x}_{k_3}\}$ from $\mathbf{V}$, such that $\mathbf{W}$ is affinely independent, then

$$M(\mathbf{x}|\mathbf{x}_0,\ldots,\mathbf{x}_{n+3})=\sum_{j=0}^{3}\lambda_j(\mathbf{x}|\mathbf{W})M(\mathbf{x}|\mathbf{V}\setminus\{\mathbf{x}_{k_j}\}), \quad (1)$$

where $\sum_{j=0}^{3}\lambda_j(\mathbf{x}|\mathbf{W})=1$ and $\sum_{j=0}^{3}\lambda_j(\mathbf{x}|\mathbf{W})\mathbf{x}_{k_j}=\mathbf{x}$.

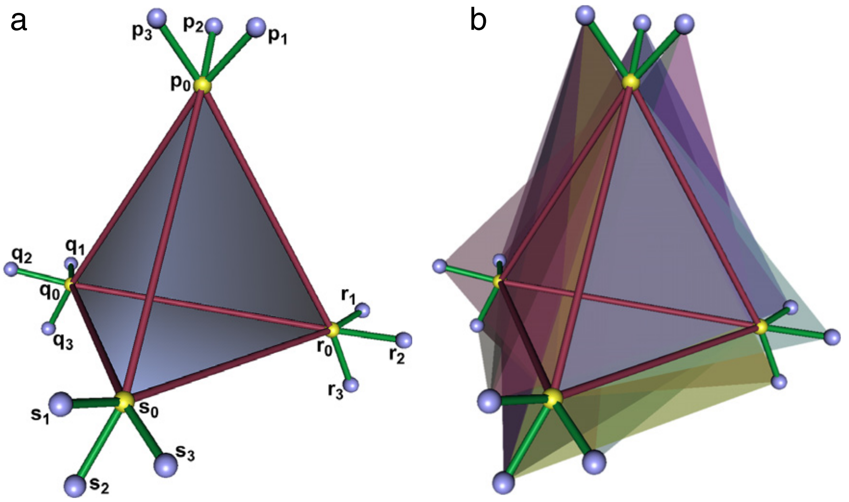

Fig. 2. (a) A domain tetrahedron demonstrated with its knot clouds assigned and labeled. The underlying tetrahedron, $(\mathbf{p}, \mathbf{q}, \mathbf{r}, \mathbf{s})$, is rendered as a shaded tetrahedron; (b) The virtual tetrahedra of the domain tetrahedron are visualized in different colors with the opacity value of 0.4. In (a) and (b), the degree of the domain is cubic hence each vertex of the tetrahedron has three sub-knots. The primary-knots are presented with yellow dots while sub-knots are depicted with blue dots. The red lines connecting the primary-knots indicate the underlying tetrahedron. Each green line here denotes the association between the primary-knot and the sub-knot. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The directional derivative of $M(\mathbf{x}|\mathbf{V})$ with respect to a vector $\mathbf{d}$ is defined as follows:

$$D_{\mathbf{d}}M(\mathbf{x}|\mathbf{V}) = n\sum_{j=0}^{3}\mu_j(\mathbf{d}|\mathbf{W})M(\mathbf{x}|\mathbf{V}\setminus\{\mathbf{x}_{k_j}\}), \qquad (2)$$

where $\mathbf{d} = \sum_{j=0}^{3}\mu_j(\mathbf{d}|\mathbf{W})\mathbf{x}_{k_j}$ and $\sum_{j=0}^{3}\mu_j(\mathbf{d}|\mathbf{W}) = 0$.

3.2. Spherical simplex spline volume

Generally, a volumetric simplex spline can take as input any domain with arbitrary geometry and topology due to its non-tensor-product nature. The spherical simplex spline volume is defined by volumetric simplex splines over a spherical volumetric domain. Here, we choose the sphere domain since mapping most organic objects in the biomedical research field to a sphere results in less distortion and more uniform distribution of sampling points, which reduces the difficulty in the fitting procedure. Note that, our volumetric simplex spline volumes represent not only boundary geometry, but also interior geometry. They can represent physical or material attributes over the entire solid as well.

3.2.1. Spherical volumetric simplex splines

Now let $\mathbb{S}^3 = \{\mathbf{x} \in \mathbb{R}^3, \|\mathbf{x}\| \leq c\}$ denote a solid sphere in $\mathbb{R}^3$. Without loss of generality, let $\mathbb{S}^3$ be a unit solid sphere, i.e., $c = 1$. Let $\mathbf{T}$ be an arbitrary "proper" tetrahedralization of $\mathbb{S}^3$. Here, "proper" means that every pair of domain tetrahedra are disjoint, or share exactly one vertex, one edge, or one face. To each vertex $\mathbf{t}$ of the tetrahedralization $\mathbf{T}$, we assign a knot cloud, which is a sequence of points $[\mathbf{t}_0, \mathbf{t}_1, \ldots, \mathbf{t}_n]$, where $\mathbf{t}_0 \equiv \mathbf{t}$. We call $\mathbf{t}$ primary-knot and $[\mathbf{t}_1, \ldots, \mathbf{t}_n]$ sub-knots. Fig. 2(a) shows 4 vertices with cubic knot clouds associated, which are labeled as $\mathbf{p}$, $\mathbf{q}$, $\mathbf{r}$, or $\mathbf{s}$ group, respectively. The primary-knots are rendered with yellow dots and sub-knots with blue dots. We will use these two colors to differentiate the primary-knots and sub-knots in the rest of the illustrations.

For every tetrahedron $I \in \mathbf{T}$, assume $I = (\mathbf{p}, \mathbf{q}, \mathbf{r}, \mathbf{s}) = (\mathbf{p}_0, \mathbf{q}_0, \mathbf{r}_0, \mathbf{s}_0)$. We call $(\mathbf{p}, \mathbf{q}, \mathbf{r}, \mathbf{s})$ the underlying tetrahedron. All the other tetrahedra $[\mathbf{p}_i, \mathbf{q}_j, \mathbf{r}_k, \mathbf{s}_l]$ with $0 < i + j + k + l \leq n$ are called the virtual tetrahedra. Fig. 2(a) shows the underlying tetrahedron with shading. Fig. 2(b) demonstrates the virtual tetrahedra rendered with different shading colors.

Then for every tetrahedron I, we require

- all the tetrahedra $[\mathbf{p}_i, \mathbf{q}_j, \mathbf{r}_k, \mathbf{s}_l]$ with $i + j + k + l \leq n$ are non-degenerate, i.e., the underlying tetrahedron and virtual tetrahedra should be valid.
- the set

$$\Omega = interior\left(\bigcap_{i+j+k+l\leq n}[\mathbf{p}_i, \mathbf{q}_j, \mathbf{r}_k, \mathbf{s}_l]\right) \qquad (3)$$

is not empty.
- if I is a boundary tetrahedron, the sub-knots assigned to the boundary vertices must lie outside of $\mathbb{S}^3$.

The condition that Ω is nonempty states that the sub-knots associated with different vertices of I are all separated from each other. The underlying tetrahedron I and its virtual tetrahedra have the same orientation. As shown in Fig. 2(a), in our framework, the orientation of the tetrahedron is defined as, observing from $\mathbf{p}_i$, the triangle formed by $(\mathbf{q}_j, \mathbf{r}_k, \mathbf{s}_l)$ is clockwise-oriented where $i + j + k + l \leq n$.

The formation of Ω can be intuitively described as: starting from the underlying tetrahedron, we chop it using the triangle faces from each virtual tetrahedron, and only keep the part inside of the triangle faces. As the starting shape of the underlying tetrahedron is convex, the chopping operation will not affect its convexity. Hence the final shape of Ω is a convex, solid polyhedron if nonempty.

Fig. 3(a) and (b) depicts the Ω of a cubic domain tetrahedron, with and without virtual tetrahedra rendered, respectively. Note that if $\Omega \neq \varnothing$, Ω must be a convex solid polyhedron formed by the interior of the underlying tetrahedron and virtual tetrahedra. Fig. 3(a) and (b) illustrate the Ω as a blue, convex, and solid polyhedron.

We then define, for each tetrahedron $I \in \mathbf{T}$ and $i+j+k+l = n$ (in the following, we use β to denote 4-tuple (i, j, k, l)), the knot sets are

$$V_\beta^I = [\mathbf{p}_0, \ldots, \mathbf{p}_i, \mathbf{q}_0, \ldots, \mathbf{q}_j, \mathbf{r}_0, \ldots, \mathbf{r}_k, \mathbf{s}_0, \ldots, \mathbf{s}_l]. \qquad (4)$$

For an example in Figs. 2 and 3, as the degree of the domain is cubic, V_β^I has 16 elements: 4 primary-knots and 12 sub-knots.

The basis functions of normalized simplex splines are then defined as

$$N_\beta^I(\mathbf{u}) = |\det(\mathbf{p}_i, \mathbf{q}_j, \mathbf{r}_k, \mathbf{s}_l)|M(\mathbf{u}|V_\beta^I). \qquad (5)$$

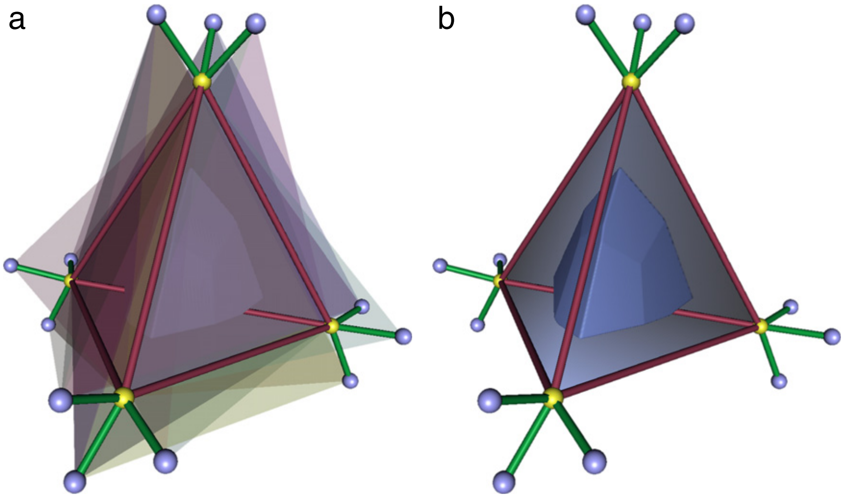

Fig. 3. (a) The Ω of a domain tetrahedron, formed by the interior of the underlying tetrahedron and virtual tetrahedra, is rendered with a blue polyhedron. The virtual tetrahedra are visualized in different colors with the opacity value 0.1; (b) The virtual tetrahedra are removed to better visualize the Ω. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

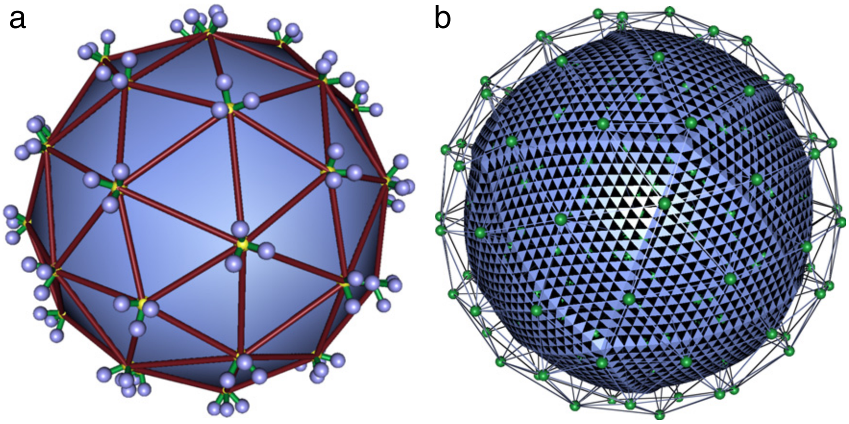

Fig. 4. (a) The spherical domain with assigned knot clouds for defining spherical volumetric simplex splines. The yellow and blue dots denote primary-knots and sub-knots, respectively; (b) The spherical simplex spline volume defined upon the domain in (a). The green dots denote the control points. The evaluated spherical volume simplex volume is scaled to show its nonempty interior property. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

These basis functions can be shown to be all non-negative and to form a partition of unity. The volumetric spherical simplex spline volume is the combination of a set of basis functions with control points $\mathbf{c}^I_\beta$:

$$\mathbf{s}(\mathbf{u}) = \sum_{I \in \mathbf{T}} \sum_{|\beta|=n} \mathbf{c}^I_\beta N^I_\beta(\mathbf{u}). \tag{6}$$

The "generalized" control points $\mathbf{c}^I_\beta$ are now $(k+3)$-dimensional vectors, including control points (p^x, p^y, p^z) for the solid geometry, and control coefficients $(g^1, \dots, g^k)$ for the attributes, where k denotes the number of attributes associated with the geometry. The spherical simplex splines are ideal to model genus-zero, heterogeneous solid objects. The number of physical properties is application-oriented. For a concise expression of the formulation, without loss of generality, we will deal with only one physical attribute in the following formulas.

Fig. 4(a) illustrates a spherical volumetric simplex spline and its domain with its cubic knot clouds associated. As observed in the figure, the sub-knots assigned to the boundary vertices of the sphere domain are positioned outside of the sphere. Fig. 4(b) shows the control space and the evaluated spherical volumetric simplex volume.

3.2.2. *Initial construction of spherical volumetric domain*

Theoretically, domain tetrahedralization, $\mathbf{T}$, can be an arbitrary tetrahedralization of a unit solid sphere, $\mathbb{S}^3$, as aforementioned in Section 3.2.1. However, in practice, two important aspects of the domain tetrahedralization should be carefully considered:

- $\mathbf{T}$ should be as uniform as possible, i.e., minimize $\frac{\max(\mathrm{Vol}_{I \in \mathbf{T}})}{\min(\mathrm{Vol}_{I' \in \mathbf{T}})}$. Uniform tetrahedralization at the same hierarchical level will decrease the recursion time while a hierarchical structure is needed.
- $\mathbf{T}$ should avoid bad-shaped tetrahedra in Delaunay tetrahedralization. Bad-shaped tetrahedra, for instance, slivers, will increase numerical error during the evaluation.

Constrained Delaunay tetrahedralization [26] can observe the second requirement, but it will introduce very large and very small tetrahedra thus cannot comply with the first requirement. Instead, we tetrahedralize a regular icosahedron and then make use of harmonic volumetric mapping to map the tetrahedralization to a solid sphere. As a result, the solid sphere tetrahedralization is uniform and its quality is better than what constrained Delaunay tetrahedralization can offer.

Fig. 5 shows the flow of domain establishment and the knots distribution. Note that, in Fig. 5(d), the sub-knots associated with

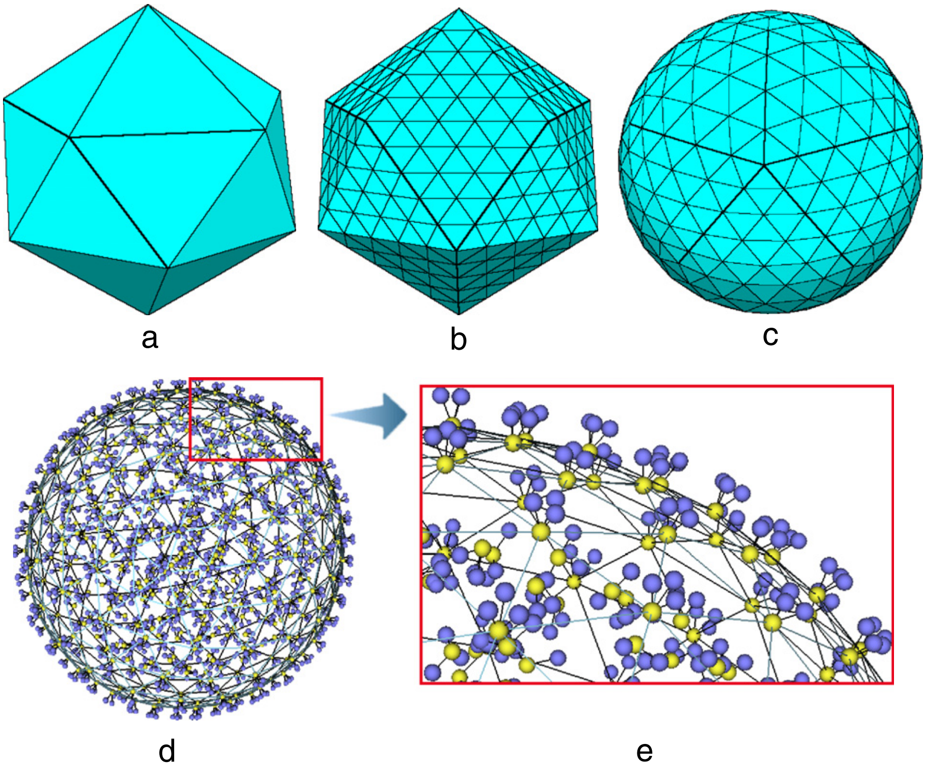

Fig. 5. (a) A regular icosahedron, which is the best approximation of a solid sphere among all regular polyhedra; (b) Tetrahedralization of (a) is uniform and it is easy to implement; (c) Harmonic mapping from (b) to a unit solid sphere yields the domain tetrahedralization, consisting of uniform and well-shaped tetrahedra; (d) A domain with cubic knot clouds assigned to (c); (e) A close view of the domain picked from (d).

boundary vertices are placed outside of the sphere. The uniform tetrahedralization may be subdivided and refined when necessary, e.g., modeling discontinuity as described later.

3.3. Model reconstruction by data fitting

Besides constructing the initial unit sphere tetrahedralization as the parametric domain, another preliminary step prior to the reconstruction of the continuous volumetric model using spherical simplex splines is to find a volumetric parameterization between the physical model and domain space.

3.3.1. Volumetric parameterization

To find a volumetric parameterization of a genus-zero solid, harmonic volumetric mapping facilitates a viable solution. Harmonic volumetric mapping was first implemented for applications by Wang et al. [27,28]. They successfully exposed its merits by applying the approach to brain mapping which can be considered as a genus-zero volume. Recently Li et al. [29] further extended the scheme to high-genus harmonic volumetric mapping and employed it in solid modeling applications. Harmonic volumetric mapping can be formulated as follows:

Given two solid objects M_1 and M_2, and their boundary surfaces ∂M_1 and ∂M_2, and suppose that $\vec{f}'$ is the conformal mapping [30,31] between ∂M_1 and ∂M_2, which is pre-computed. The harmonic volumetric mapping $\vec{f} : M_1 \longmapsto M_2$ satisfies:

$$\begin{cases} \nabla^2 \vec{f}(v) = 0, & v \in M_1 \setminus \partial M_1, \\ \vec{f}(v) = \vec{f}'(v), & v \in \partial M_1, \end{cases}$$

where the ∇^2 is the Laplacian operator defined continuously in 3D as

$$\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2},$$

and $\nabla^2 \vec{f} = 0$ for $\vec{f} = (f_0, f_1, f_2)$ is equivalent to $\nabla^2 f_i = 0$ for all $i = 0, 1, 2$.

The harmonic volumetric mapping f here minimizes a harmonic energy $E(f)$ [28], which is defined as

$$E(f) = \sum_{(u,v)} k(u, v)(f(u) - f(v))^2, \qquad (7)$$

where $k(u, v)$ is the string constant defined in the edge between u and v. Here, f can be solved using a steepest descent algorithm.

The algorithmic procedure of harmonic volumetric mapping is concisely summarized as follows:

(1) For each boundary vertex, v, $v \in \partial M_1$, let $\vec{f}(v) = \vec{f}'(v)$; for each interior vertex, v, $v \in M_1 \setminus \partial M_1$, let $\vec{f}(v) = \vec{0}$, compute the harmonic energy E_0 using Eq. (7).
(2) For each interior vertex, v, $v \in M_1 \setminus \partial M_1$, compute its derivative $D\vec{f}$ using a steepest descent algorithm, then update $\vec{f}(v)$ by $\delta \vec{f}(v) = -D\vec{f}(t)\delta t$, δt is the step length.
(3) Compute the harmonic energy E; if $E - E_0$ is less than user specified threshold δE, the algorithm stops; Otherwise assign E to E_0 and repeat step (2) through step (3).

Fig. 6 shows the harmonic volumetric mapping from one brain to a solid unit sphere. After the mapping has been established, the point parameterization and correspondence between the domain and the object can now be stored as the input of our spherical simplex spline model reconstruction algorithm.

3.3.2. Fitting with spherical volumetric simplex splines

After harmonic volumetric mapping, a finite number of discretized sampling points of the physical object, $(x_i, y_i, z_i, \rho_i)_{i=1}^m$, and their parametric coordinates in the domain, $(u_i, v_i, w_i)_{i=1}^m$, can be retrieved. ρ_i denote a physical attribute. Note that, there could be multimodality physical attributes with more dimensions. In this

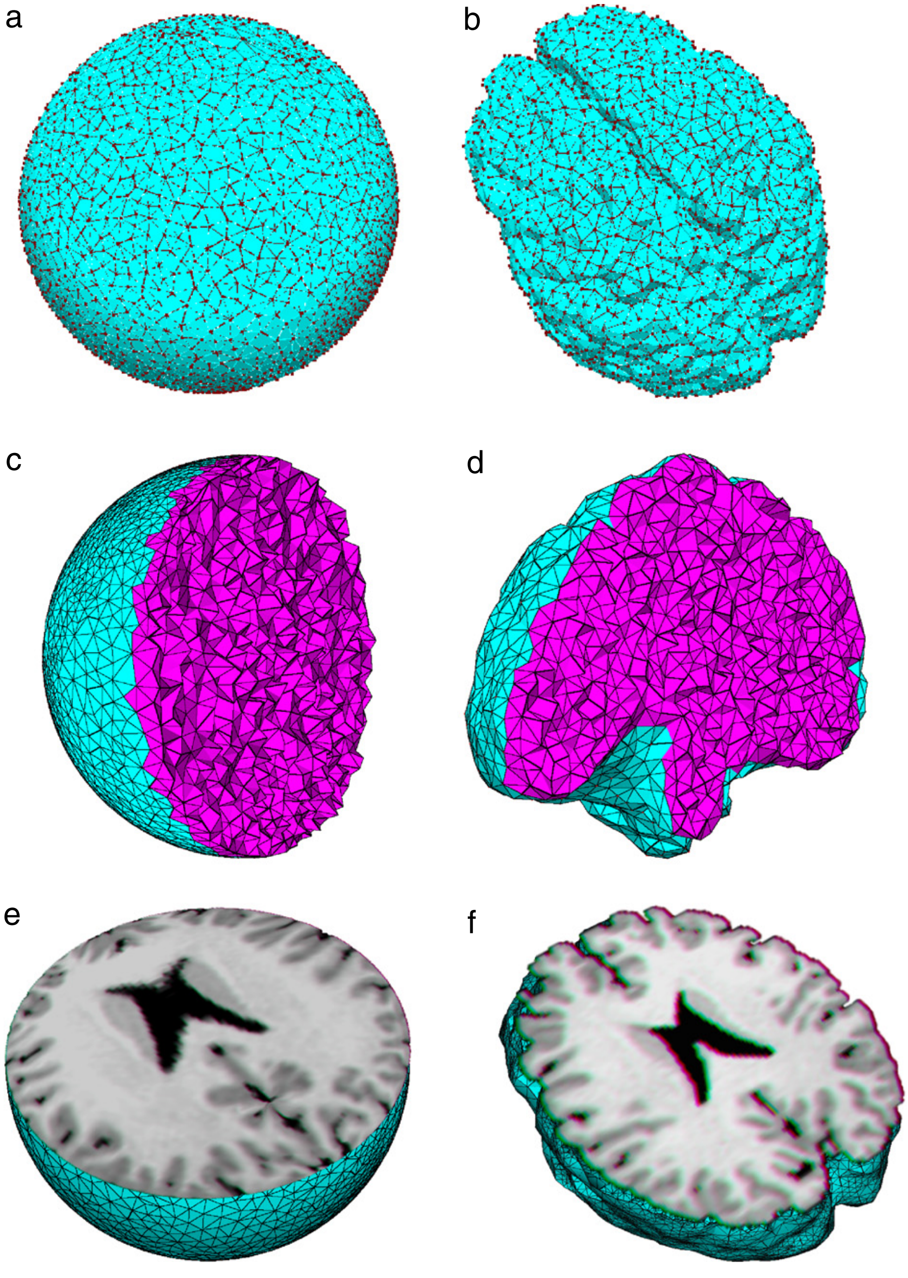

Fig. 6. (a) The discretized point set in the spherical domain space; (b) The discretized data point set in the physical space, from the same angle of view as (a); (c–f) The shapes are cut into halves sagittally (c–d) and axially (e–f) in order to show the interior mapping between the parametric domain and the physical object.

case, all we need to do is to increase the dimensions and add the additional variables into the sampling. All the computation remains the same. Without loss of generality, we only consider one type of attribute here in order to simplify the mathematical notation. The sampling point pairs indicates the parameterization from the solid sphere domain to the to-be-modeled object. Volumetric simplex spline is an ideal tool for fitting the geometry as well as the physical properties of the volumetric object. In this section, we will describe how to fit spherical volumetric simplex splines to the real-world model.

The problem of model reconstruction in our system can be stated as follows: given a set $P = \{\mathbf{p}_i\}_{i=1}^m$ of points, $\mathbf{p}_i = (x_i, y_i, z_i, \rho_i) \in \mathbb{R}^4$, and $G = \{\mathbf{g}_i\}_{i=1}^m$, $\mathbf{g}_i = (x_i, y_i, z_i) \in \mathbb{R}^3$ denoting the pure geometry extracted from the sampling points, find a volumetric simplex splines volume $\mathbf{s} : \mathbb{R}^3 \to \mathbb{R}^3$ that approximates G.

Since we are interested in reconstructing the model with respect to its solid geometry, our spherical simplex spline volumes are vector functions, i.e., the control points $\mathbf{c}_\beta^I \in \mathbb{R}^3$ are vectors. Unlike the existing fitting algorithms with simplex splines which usually find the parametric domain which is close to the original geometry of the to-be-fitted dataset [1,11], we use the position (u_i, v_i, w_i) within the solid sphere as the data point $\mathbf{g}_i$'s parametric value. Therefore, we need to minimize the following objective function:

$$\min E_{\text{dist}}(\mathbf{s}) = \sum_{i=1}^{m} (\mathbf{g}_i - \mathbf{s}(u_i, v_i, w_i))^2. \tag{8}$$

Eq. (8) is a typical least squares problem. If the control points are treated as free variables, it falls into a very special category of nonlinear programming, i.e., unconstrained convex quadratic programming, which has the following form:

$$E_{\text{dist}} = \frac{1}{2}\mathbf{x}^{\mathrm{T}} Q \mathbf{x} + \mathbf{c}^{\mathrm{T}}\mathbf{x} + f,$$

where $\mathbf{x} = (\ldots, \mathbf{c}_\beta^I, \ldots)^{\mathrm{T}}$,

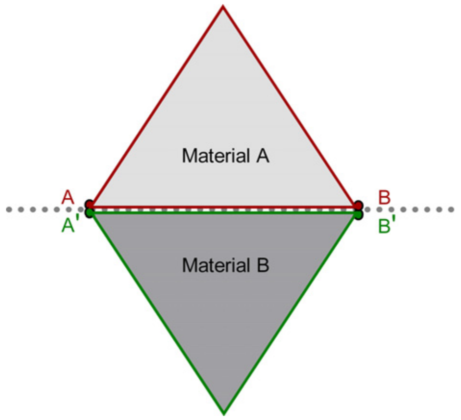

Fig. 7. Modeling discontinuities with separated domain triangles. Even though A and A′ are co-located, and B and B′ are co-located, the domain triangles in red and green belong to two different domains. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$Q = \begin{pmatrix} & \vdots & \\ \cdots & 2\sum_{i=1}^{m} N^{I}_{\beta}(u_i, v_i, w_i) N^{I'}_{\beta'}(u_i, v_i, w_i) & \cdots \\ & \vdots & \end{pmatrix},$$

$$\mathbf{c} = \left(\ldots, -2\sum_{i=1}^{m} \mathbf{g}_i N^{I}_{\beta}(u_i, v_i, w_i), \ldots\right)^{\mathrm{T}},$$

and $f = \sum_{i=1}^{m} \mathbf{g}_i^2$. Note that, Q is a positive definite, symmetric and sparse matrix. The interior-point method can solve this problem very efficiently.

After the reconstruction procedure, we can achieve an integrated representation incorporating the object's solid geometry, **s**, and its material attribute, **d**, at the same time. The scheme can be expressed as

$$\begin{bmatrix} \mathbf{s} \\ \mathbf{d} \end{bmatrix} (\mathbf{u}) = \sum_{I \in \mathbf{T}} \sum_{|\beta|=n} \begin{bmatrix} \mathbf{c} \\ d_c \end{bmatrix} N(\mathbf{u}|V^{I}_{\beta}), \tag{9}$$

where **c** and d_c are the control points and control coefficients for solid geometry and material attributes, respectively.

To model discontinuity in an attribute field, we first detect where the discontinuity occurs, then decompose the original domain into two separated new domains with shared vertices and edges as the 2D illustration in Fig. 7. This simple mechanism maintains the consistent structure of the domains. The evaluation, hierarchy structure, and data structure all remain the same. Therefore, we can perform the same evaluation on these two domains simultaneously as if the evaluation were performed on a single domain. With the association of different control coefficients, the functional evaluation can output a discontinuity in material field corresponding to the shared edges. This change will not affect the geometry of the DSVSS volume as long as the associated control points remain the same.

3.4. *Dynamic spherical simplex splines*

In this section, we formulate our dynamic spherical volumetric simplex splines. We integrate mass, dissipation, and deformation energy into static simplex spline models, and employ Lagrangian dynamics to derive their equations of motion. Consequently, the static control points of the geometric model become generalized time-varying physical coordinates in the dynamic model.

3.4.1. *Geometry and kinematics of simplex spline volumes*

The dynamic simplex splines further extend the geometric simplex splines by incorporating time into the volume representation. Now the function of representation bears both parametric variable **u** and time t as follows:

$$\mathbf{s}(\mathbf{u}, t) = \sum_{I \in \mathbf{T}} \sum_{|\beta|=n} \mathbf{c}^{I}_{\beta}(t) N^{I}_{\beta}(\mathbf{u}). \tag{10}$$

For simplicity of formulation expression, we define the vector of generalized coordinates of control points $\mathbf{c}^{I}_{\beta}$ as:

$$\mathbf{c} = [\ldots, {\mathbf{c}^{I}_{\beta}}^{\top}, \ldots]^{\top}, \tag{11}$$

where $\top$ denotes transposition. We then express Eq. (10) as $\mathbf{s}(\mathbf{u}, \mathbf{c})$ in order to emphasize its dependence on **c** whose components are functions of time. Hence, the velocity of the dynamic simplex splines is:

$$\dot{\mathbf{s}}(\mathbf{u}, t) = \mathbf{J}\dot{\mathbf{c}}, \tag{12}$$

where the overstruck dot denotes a time derivative and Jacobian matrix $\mathbf{J}(\mathbf{u})$ is the concatenation of the vectors $\partial\mathbf{s}/\partial\mathbf{c}^{I}_{\beta}$. Assuming there is m tetrahedra in the parametric domain, β traverses $k = (n+1)(n+2)(n+3)/6$ possible tetrads whose components sum to n. Because **s** is a 4-vector and **c** is an $M = 4mk$ dimensional vector, **J** is a $4 \times M$ matrix, which is expressed as

$$\mathbf{J} = \begin{bmatrix} \ldots, \begin{bmatrix} N^{I}_{\beta} & 0 & 0 & 0 \\ 0 & N^{I}_{\beta} & 0 & 0 \\ 0 & 0 & N^{I}_{\beta} & 0 \\ 0 & 0 & 0 & N^{I}_{\beta} \end{bmatrix}, \ldots \end{bmatrix}, \tag{13}$$

where $N^{I}_{\beta}(\mathbf{u}) = \frac{\partial \mathbf{s}_x}{\partial \mathbf{c}^{I}_{\beta_x}} = \frac{\partial \mathbf{s}_y}{\partial \mathbf{c}^{I}_{\beta_y}} = \frac{\partial \mathbf{s}_z}{\partial \mathbf{c}^{I}_{\beta_z}} = \frac{\partial \mathbf{s}_d}{\partial \mathbf{c}^{I}_{\beta_d}}$.

The subscripts x, y, z and d denote derivatives of the components of the 4-vector: Cartesian coordinates and physical property, respectively. Apparently, the solid volume can be presented as the production of the product of the Jacobian matrix and the generalized coordinate vector,

$$\mathbf{s}(\mathbf{u}, \mathbf{c}) = \mathbf{J}\mathbf{c}. \tag{14}$$

3.4.2. *Lagrange equations of motion*

Lagrange dynamics are widely used in physics-based shape design. In this section, we derive the equations of motion of dynamic simplex splines by applying Lagrangian dynamics [32]. We express the kinetic energy due to the prescribed mass distribution function $\boldsymbol{\mu}(u, v, w)$, and a Raleigh dissipation energy due to a damping density function $\boldsymbol{\gamma}(u, v, w)$. Both energy functions are defined over the parametric domain of the volume. The mass distribution function and damping density function are reconstructed with spherical volumetric simplex splines as well, as described in Section 3.3.2. A 3D thin-plate-like energy under tension energy model [14,33,16,34] is employed here in order to define an elastic potential energy,

$$U = \frac{1}{2} \iiint (\boldsymbol{\alpha}_{1,1}\mathbf{s}_u^2 + \boldsymbol{\alpha}_{2,2}\mathbf{s}_v^2 + \boldsymbol{\alpha}_{3,3}\mathbf{s}_w^2 + \boldsymbol{\beta}_{1,1}\mathbf{s}_{uu}^2 + \boldsymbol{\beta}_{1,2}\mathbf{s}_{uv}^2 + \boldsymbol{\beta}_{1,3}\mathbf{s}_{uw}^2 + \boldsymbol{\beta}_{2,2}\mathbf{s}_{vv}^2 + \boldsymbol{\beta}_{2,3}\mathbf{s}_{vw}^2 + \boldsymbol{\beta}_{3,3}\mathbf{s}_{ww}^2)\mathrm{d}u\mathrm{d}v\mathrm{d}w. \tag{15}$$

The subscripts on **s** denote the parametric partial derivatives. The $\boldsymbol{\alpha}_{i,j}(u, v, w)$ and $\boldsymbol{\beta}_{i,j}(u, v, w)$ are elasticity functions which control tension and rigidity, respectively. Other energies, requiring greater computational cost, are also applicable, for instance, the non-quadratic, curvature-based energies in [35,36]. Applying the Lagrangian formulation, we obtain the second-order equations of motion

$$\mathbf{M}\ddot{\mathbf{c}} + \mathbf{D}\dot{\mathbf{c}} + \mathbf{K}\mathbf{c} = \mathbf{f}_c, \tag{16}$$

where the mass matrix is

$$\mathbf{M} = \iiint \boldsymbol{\mu} \mathbf{J}^{\top}\mathbf{J} \mathrm{d}u\mathrm{d}v\mathrm{d}w, \tag{17}$$

the damping matrix is

$$\mathbf{D} = \iiint \boldsymbol{\gamma} \mathbf{J}^{\top}\mathbf{J} \mathrm{d}u\mathrm{d}v\mathrm{d}w, \tag{18}$$

and the stiffness matrix is

$$\begin{aligned}\mathbf{K} = \iiint & (\boldsymbol{\alpha}_{1,1}\mathbf{J}_u^{\top}\mathbf{J}_u + \boldsymbol{\alpha}_{2,2}\mathbf{J}_v^{\top}\mathbf{J}_v + \boldsymbol{\alpha}_{3,3}\mathbf{J}_w^{\top}\mathbf{J}_w \\ & + \boldsymbol{\beta}_{1,1}\mathbf{J}_{uu}^{\top}\mathbf{J}_{uu} + \boldsymbol{\beta}_{1,2}\mathbf{J}_{uv}^{\top}\mathbf{J}_{uv} + \boldsymbol{\beta}_{1,3}\mathbf{J}_{uw}^{\top}\mathbf{J}_{uw} \\ & + \boldsymbol{\beta}_{2,2}\mathbf{J}_{vv}^{\top}\mathbf{J}_{vv} + \boldsymbol{\beta}_{2,3}\mathbf{J}_{vw}^{\top}\mathbf{J}_{vw} + \boldsymbol{\beta}_{3,3}\mathbf{J}_{ww}^{\top}\mathbf{J}_{ww})\mathrm{d}u\mathrm{d}v\mathrm{d}w.\end{aligned} \tag{19}$$

M, **D** and **K** are all $M \times M$ matrices. $\mathbf{f}_c$ is the generalized force, which is obtained through the principle of virtual work [32] done by the applied force distribution $\mathbf{f}(u, v, w, t)$. $\mathbf{f}_c$ can be computed as follows:

$$\mathbf{f}_c = \iiint \mathbf{J}^{\top}\mathbf{f}(u, v, w, t)\mathrm{d}u\mathrm{d}v\mathrm{d}w. \tag{20}$$

4. Finite element framework

The evolution of the vector of generalized coordinates, $\mathbf{c}(t)$, is determined by the second-order nonlinear differential equation. Eq. (16) with physical parameter dependent matrices, does not have an analytical solution. Instead, we obtain an efficient numerical implementation using finite-element techniques.

Standard finite element methods explicitly integrate the individual element matrices into the global matrices that appear in the discrete equations of motion [37]. Although applicable in some environments, it is infeasible in our infrastructure because of its unacceptably high computational cost. Instead, we pursue an iterative matrix solver to avoid the cost of assembling the global matrices **M**, **D**, and **K**, working instead with the individual dynamic simplex spline element matrices. We construct finite element data structures, similar to [19], which facilitates the parallel computation of element matrices.

4.1. Data structures for dynamic simplex spline finite elements

We define an element data structure which contains the geometric specification of the tetrahedron patch element along with its physical properties. In each element, we allocate an elemental mass, damping, and stiffness matrix, and include the quantities such as the mass $\boldsymbol{\mu}(u, v, w)$, damping $\boldsymbol{\gamma}(u, v, w)$, and elasticity $\boldsymbol{\alpha}_{i,j}(u, v, w)$ and $\boldsymbol{\beta}_{i,j}(u, v, w)$ functions. A complete dynamic simplex spline consists of an ordered array of elements with additional information. The element structure includes pointers to appropriate components of the global vector **c**. Neighboring tetrahedra will share some generalized coordinates.

The physical parameters, such as mass $\boldsymbol{\mu}(u, v, w)$, damping $\boldsymbol{\gamma}(u, v, w)$, and elasticity, $\boldsymbol{\alpha}_{i,j}(u, v, w)$ and $\boldsymbol{\beta}_{i,j}(u, v, w)$, need to be measured and computed before the calculation of element matrices. In this paper, as the goal of the applications is to simulate the biomechanical behavior of the brain, we directly adopt $\boldsymbol{\mu}$ and $\boldsymbol{\gamma}$ from the brain study conducted by Zhang et al. [38]. According to the relationship of elastic moduli of elastic isotropic materials [39], $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ can be computed from Bulk modulus and Poisson's ratio as follows:

$$\boldsymbol{\alpha} = 3B(1 - 2\upsilon), \tag{21}$$

$$\boldsymbol{\beta} = \frac{3B(1 - 2\upsilon)}{(2 + 2\upsilon)}, \tag{22}$$

where B is the Bulk modulus and υ is the Poisson's ratio of brain tissue. After we get the physical parameters for different types of brain tissues, we then take these coefficients into the fitting procedure to integrate them into our DSVSS framework. Now the volume representation can be described as follows:

$$\begin{bmatrix} \mathbf{s} \\ \boldsymbol{\mu} \\ \boldsymbol{\gamma} \\ \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} (\mathbf{u}) = \sum_{I \in \mathbf{T}} \sum_{|\beta|=n} \begin{bmatrix} \mathbf{c} \\ \mu_c \\ \gamma_c \\ \alpha_c \\ \beta_c \end{bmatrix} N(\mathbf{u}|V_{\beta}^{I}), \tag{23}$$

where **c** and $\mu_c, \gamma_c, \alpha_c, \beta_c$ are the control points and control coefficients for solid geometry and material physical attributes, respectively. Homogeneously taking the parameters into the element without fitting may sound feasible. However, it is the fitting procedure that takes attribute field discontinuity into account to achieve a model of high fidelity.

4.2. Calculation of element matrices

We employ Gaussian quadrature [40] to numerically evaluate the integral expressions for the mass, damping, and stiffness matrices associated with each element. In this section, we explain the expression of the element damping matrix in detail; the expressions of mass and stiffness matrix will follow suit. Assuming the parametric domain of the element is $I(v_0, v_1, v_2, v_3)$ where v_i denotes the vertex, the expression for entry d_{ij} of the damping matrix takes the integral form

$$d_{ij} = \int_{I \in T} \int_{I(v_0,v_1,v_2,v_3)} \boldsymbol{\gamma}(u, v, w) f_{ij}(u, v, w)\mathrm{d}u\mathrm{d}v\mathrm{d}w, \tag{24}$$

where f_{ij} is evaluated using the recursive expression in Eq. (1). Given integers N_g, we can find the corresponding Gauss weights a_g, and parametric abscissas u_g, v_g, and w_g such that d_{ij} can be approximated by

$$d_{ij} \approx \sum_{g=1}^{N_g} a_g \boldsymbol{\gamma}(u_g, v_g, w_g) f_{ij}(u_g, v_g, w_g). \tag{25}$$

In our system, we choose N_g to be 10 for cubic dynamic simplex splines. Because of the irregularity of the knot distribution, many of the f_{ij} vanish over the sub-space of $I(v_0, v_1, v_2, v_3)$. We can further subdivide the $I(v_0, v_1, v_2, v_3)$ to minimize the numerical error.

4.3. Discrete dynamics equations

In this section, we will derive the discrete dynamics equations based on Eq. (16). In order to integrate it in a simulation system, e.g., tissue simulation during surgery, it is important to provide users with visual feedback about the evolution state of the DSVSS model. Rather than using computation-intensive time integration methods which may traverse the largest possible time steps, it is more crucial to provide a smoothly simulated display by maintaining the continuity of the dynamics form one step to the next. Therefore, it is very desirable to employ less costly yet stable time integration methods that take reasonable time steps.

The state of the dynamic simplex splines at time $t + \Delta t$ is integrated using prior states at t and $t - \Delta t$. To maintain the stability of the integration scheme, especially for high stiffness configurations with large elasticity functions, we use an implicit time integration method, which employs discrete derivatives of **c** using backward differences. The velocity expression is

$$\dot{\mathbf{c}}^{t+\Delta t} \approx (\mathbf{c}^{(t+\Delta t)} - \mathbf{c}^{(t-\Delta t)})/2\Delta t \tag{26}$$

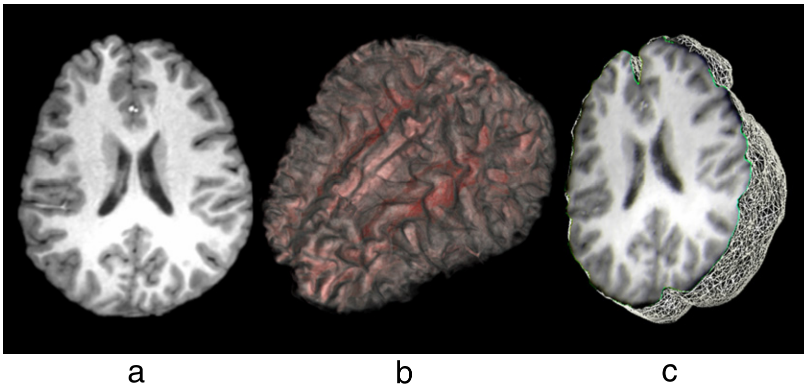

Fig. 8. (a) An axial view of a high-resolution brain SPGR MRI dataset; (b) Volume visualization of the reconstructed DSVSS volume; (c) The volume is split to show its reconstructed interior intensities.

and the acceleration expression is

$$\ddot{\mathbf{c}}^{t+\Delta t} \approx (\mathbf{c}^{(t+\Delta t)} - 2\mathbf{c}^{(t)} + \mathbf{c}^{(t-\Delta t)})/\Delta t^2. \tag{27}$$

Then the time integration formula can be expressed as

$$\begin{aligned}&(2\mathbf{M} + \Delta t\mathbf{D} + 2\Delta t^2\mathbf{K})\mathbf{c}^{(t+\Delta t)}\\ &\quad = 2\Delta t^2\mathbf{f}_c + 4\mathbf{M}\mathbf{c}^{(t)} - (2\mathbf{M} - \Delta t\mathbf{D})\mathbf{c}^{(t-\Delta t)}\end{aligned} \tag{28}$$

where the superscripts denote evaluation of the quantities at the indicated times. The matrices and forces are evaluated at time t. Our extensive experiments have shown that this discretization scheme produces satisfactory results. Instability due to large transient applied forces can be reduced by shortening the time integration step adaptively.

The equations of motion allow a physically realistic simulation of real-world models with complex dynamics. However, it is possible to make simplifications to the equations of motion to further reduce the computational cost of solving Eq. (28) when we simulate some more complicated volumes which bear more tetrahedra in its domain. In certain solid modeling and simulation applications where the inertial terms are not taken into count, the Eq. (16) can be simplified by setting the mass density function to zero. Without computation of the acceleration terms or storage of mass matrices, the algorithm is more efficient. With zero mass density, Eq. (16) simplifies to

$$\mathbf{D}\dot{\mathbf{c}} + \mathbf{K}\mathbf{c} = \mathbf{f}_c. \tag{29}$$

Discretizing the corresponding derivatives of $\mathbf{c}$ in Eq. (29) with backward differences, the integration formula becomes

$$(\mathbf{D} + \Delta t\mathbf{K})\mathbf{c}^{(t+\Delta t)} = \Delta t\mathbf{f}_c + \mathbf{D}\mathbf{c}^{(t)}. \tag{30}$$

5. Brain simulation using DSVSS volume

With the reconstruction of brain model from both MRI data and material map using our spherical volumetric simplex splines, we can obtain an analytic representation simultaneously describing both geometric and physical properties of the brain. Thus, brain simulations, such as brain shifting, deformation, and brain injury predication, can be achieved via the simulation-based analysis. In this section, we present the accurate brain reconstruction and simulation using our unified scheme, DSVSS volume. The reconstruction process is fully automated, and for brain simulation, the user only needs to initialize a few environmental parameters, e.g., the gravity and the resected skull in the brain shifting simulation.

5.1. Fitting spherical volumetric simplex splines to brain data

Taking a set of high-resolution brain SPGR MR scans, we first strip away the skull and only retain the brain volume as shown in Fig. 8(a). With the initial tetrahedralization of the brain model and harmonic volumetric mapping, we can obtain the parameterization of the data points of the brain tetrahedralization as described in Section 3.3, i.e., the parameterization describes the correspondence between the brain data points and parametric coordinates in the sphere domain. Fitting spherical volumetric simplex splines to the geometric representation, we can reconstruct the geometry of the brain nicely as shown in Fig. 6. To model the intensities (for visualization purpose) and material distribution (for simulation purpose), we can start with the same spherical tetrahedral domain, and then subdivide and refine the domain [12], when necessary, to model more sophisticated material variations or discontinuities as described in Section 3.3.2. Note that, the intensities and material of brain structures are related since the imaging procedure can be considered as a function mapping of the material maps to the scanned images. So the required domain for intensities and material distributions are very similar. Fig. 8 shows the reconstruction result with different rendering techniques and Eq. (31) shows the reconstructed representation,

$$\begin{bmatrix}\mathbf{s}\\ \mathbf{d}\\ \mathbf{I}\end{bmatrix}(\mathbf{u}) = \sum_{I\in\mathbf{T}}\sum_{|\beta|=n}\begin{bmatrix}\mathbf{c}\\ d_c\\ I_c\end{bmatrix}N(\mathbf{u}|V_\beta^I), \tag{31}$$

where $\mathbf{s}$ denotes the solid geometry of the brain, $\mathbf{d}$ denotes the reconstructed physical attributes of the brain, and $\mathbf{I}$ denotes the reconstructed image intensities from the high-resolution SPGR MRI sequence. $\mathbf{c}$, d_c and I_c are the control points and control coefficients. The accuracy of the data fitting is documented in the experimental result section. After obtaining a high-quality DSVSS volume representation of the brain model, we can use it to simulate brain deformation during surgery for computer-assisted surgical planning/surgery, or even for an innovative simulation-based diagnosis for brain injury under blunt impact.

5.2. Brain shifting during surgery

As is known by brain surgery professionals, after a patient's skull is open, the brain will behave with increasing deformation, known as brain shifting, during ongoing surgical procedures, predominantly due to gravity and the drainage of cerebrospinal fluid. This will inevitably lead to the repositioning of the surgical targets embedded in brain. As a compensation to increase the spatial accuracy of modern neuronavigation systems, intraoperative magnetic resonance imaging (IMRI) is widely used for quantitative

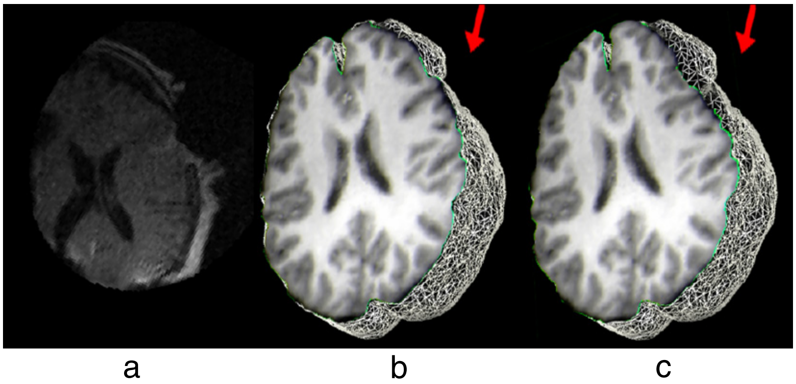

Fig. 9. (a) One slice view of IMRI image; (b) The reconstructed DSVSS volume, where the cross-sectional view displays the DSVSS-captured image intensities reconstructed from the pre-operative high-resolution SPGR images; (c) The brain deformation simulated using our system, where the cross-sectional view is captured, from the same view angle as (b), to show the displacement from (b), and the green contour indicates the extent of displacement at the boundary. In (b) and (c) the red arrow denotes the orientation of gravity, and its position denotes the resected skull. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

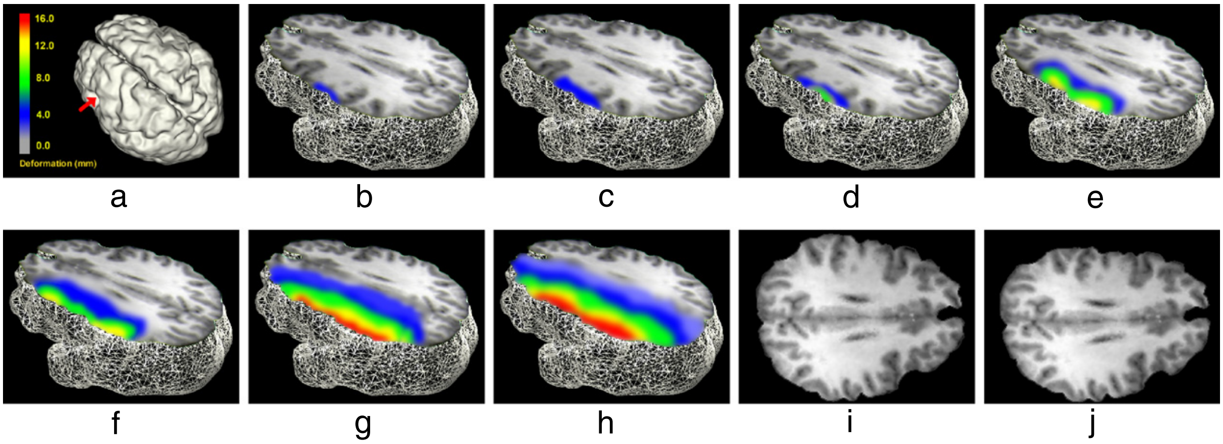

Fig. 10. (a) The color map used to describe the deformation scale. The red arrow on the ISO-surface indicates the position where the skull is resected; (b–h) Brain shifting simulation with a time interval of 75 ms; (i–j) To better visualize the deformation, cross-section views of the first key frame (b) and last one (h) are retrieved. Deformed junction between the two hemispheres indicates the global brain shifting. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

analysis and visualization of this phenomenon [41]. Nevertheless, despite its virtually real-time aspects, IMRI only provides very low-resolution intraoperative MR image which can never substitute the high-resolution pre-operative SPGR MR image used to determine with high accuracy key dimensions of the brain and the locations of the surgical targets embedded in the brain. We employ our dynamic spherical volumetric simplex splines model into the brain simulation to compute brain shifting.

In our framework, brain shifting can be simulated by applying a constant gravity force $\vec{G}$ to the brain. The material properties that we used in our experiments were obtained from *the biomechanics group* at Wayne State University (WSU). After setting up the physical parameters of an individual brain, we also need to take the natural boundary of the brain, the skull, into consideration. The fact is that no matter how the brain behaves due to deformation, it lies inside the skull, i.e., its natural boundary will not exceed the skull. Therefore, spatial geometric constraints need to be enforced. We add the soft constraints with forces. When there is shifting outside the boundary, we insert corresponding forces along the opposite direction of the movement to the simulation procedure.

Fig. 9 illustrates the brain shifting simulation using our framework when taking out the resected skull over the right temporal lobe. The green contour shows the deformation clearly. Our shifting simulation results agree highly with the fact captured by IMRI. The experiments show that it is effective to use our model to recover motion and deformation from image data. Based on 20 simulation experiments, quantitative comparisons between the IMRI volumes and our simulated brain volumes by co-registration show that our system can achieve an excellent accuracy of 92.2%. The accuracy of a single simulation, denoted by A, is calculated as the normalized sum of squared differences between the two volumes,

$$A = 1 - \frac{\sum_a \|S - R\|^2}{\sum_a \|R\|^2}, \tag{32}$$

where S is the volume obtained from our shifting simulation results and R is the registered IMRI volume. To make the comparison substantial and intra-sequence, we first register the MRI volume to the IMRI volume. Fig. 10 depicts another brain shifting simulation. The skull is resected over the left temporal lobe. The color map is blended into the figure to better visualize the deformation scale. Note that, when surgical tools are operating in the brain, there will be larger shifting and deformation.

As demonstrated from the available comparison and evaluation, our framework can accurately simulate the deformation of the brain (e.g., $\mathbf{s}(t)$) and simultaneously present high-quality and high-resolution visualization using the transformed SPGR image intensities, $\mathbf{I}$, modeled in the reconstructed simplex spline volume

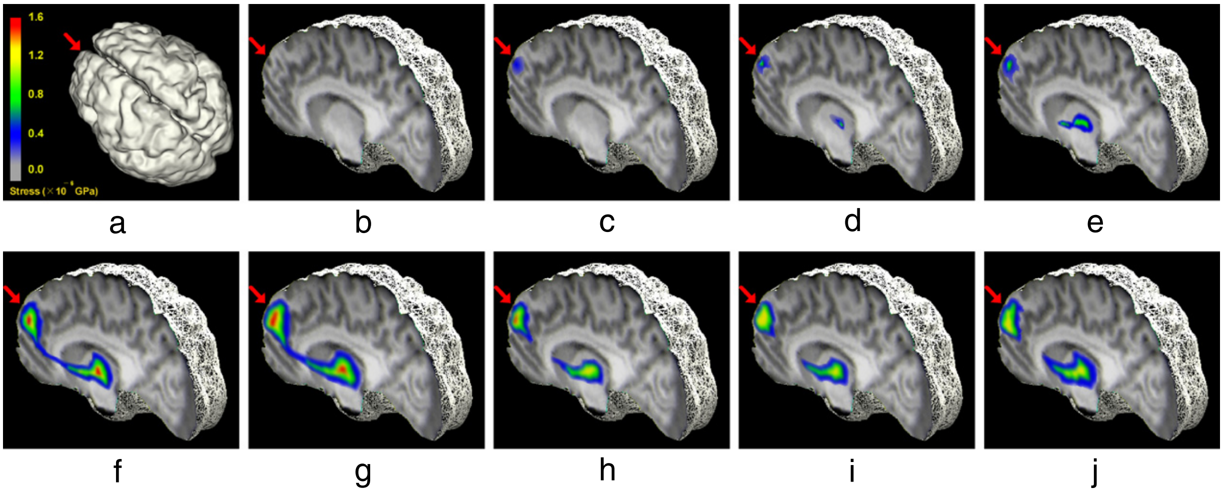

Fig. 11. (a) The color map used to describe the stress field. The red arrow on the ISO-surface indicates the position where the blunt impact occurs; (b–j) Brain injury simulation with a time interval of 3 ms. The blunt impact occurs at the front lobe. Simulation results indicate that in addition to the spot directly under the impact, there are some other positions where bleeding may happen. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

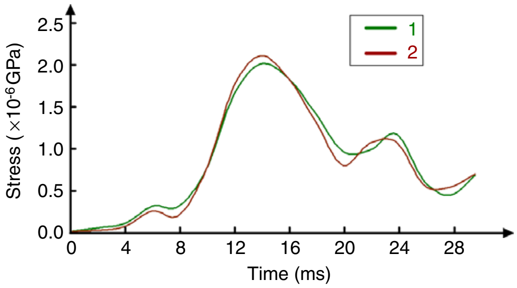

Fig. 12. Comparison of stress evolutions of the right thalamus under a blunt impact. The green one is the simulation curve obtained from the real biomechanic experiments and the red one is the result simulated using our framework. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

(see Eq. (31)). It is very promising to use the framework in both surgical planning (e.g., predicting the shifting of the targets) and computer-assisted surgery (e.g., repositioning the targets with high-resolution display, **I**, automatically computed based on the realistic deformation of the reconstructed brain, $\mathbf{s}(t)$).

5.3. *Brain injury prediction*

Here, we refer the brain injury prediction as a procedure of finding out the extent and location of an injury in the brain during a blunt impact. This injury frequently occurs to automobile drivers during a collision and sports players during acute sports activities such as football. Current brain surgeons and professionals rely indispensably on these modern neuroimaging and neuronavigation systems to pinpoint the injury. Clinically, the identification of the site and extent of an injury within the brain without subjecting the patient to imaging scanning, has its advantages. For instance, head injured patients are difficult to control and may not remain still long enough for the completion of the scanning. In some severe cases, time is so limited that patients cannot even afford such a pre-operative scanning. Thus the demand of simulation-based Computer Aided Diagnosis (CAD) solution goes up to high gear. Often, the solution is referred as "brain injury modeling".

One critical issue about the BIM technique is to derive a patient-specific brain model based on a template model, thus skipping neuroimaging and neuronavigation, and saving computational time as well as pre-operation time. One widely employed way is to modify the exterior surface of each substructure from a general brain model followed by re-generation of the mesh. Ferrant et al. [42] and Miga et al. [43] developed their approaches respectively using this approach by meshing the entire brain without considering anatomical structures and material differences. Obviously, this approach is not accurate since the brain geometry, structures, and heterogeneous material variations are not considered. We employ our dynamic spherical simplex splines-based simulation framework to handle the situation. As for developing a patient-specific model, our method can quickly modify the control points/coefficients according to the data fitting of the available data or information of the patient.

In our framework, we compute the stress field of the human brain under blunt impact using our DSVSS volume. The human brain has highly heterogenous physical properties in different areas of the brain, such as the white matter, the gray matter, the cerebellum, the brainstem, the lateral ventricles, the third ventricles, the bridge veins, and so on. From this perspective, brain structures under direct impact are not necessary the parts where brain injuries occur. With our unified solid representation through dynamic spherical volumetric simplex splines, blunt-impact injury can be simulated using our framework by applying an instantaneous impact to the brain model under given approximate impact conditions. The model incorporated in our framework cannot only assist the physician in identifying the location and extent of the damaged area without pre-operative scanning but also enable the designer of automobiles and helmets to improve the human-centered design of head-protective facilities.

Fig. 11 demonstrates a brain injury prediction with a blunt impact on the frontal lobe. Time interval here is 3 ms. Note that, we assume that the brain always lies inside the skull during the simulation. As in a brain shifting simulation, we add corresponding contacting forces into the simulation when the brain is shifting outside the boundary. The corresponding contacting forces is along the opposite direction and linear to the extent of the brain movement. Fig. 11(b–j) shows the stress fields of the brain in each time step. The redder area indicates higher stress, which is a sign for a higher possibility of injury and bleeding. In the figure, the thalamus is under bigger stress as well beside the place under the direct blunt impact. The result complies with the ground truth captured from the real biomechanic experiments on a human corpus model. Quantitative evaluation of our simulation result is obtained through a comparison with the ground truth.

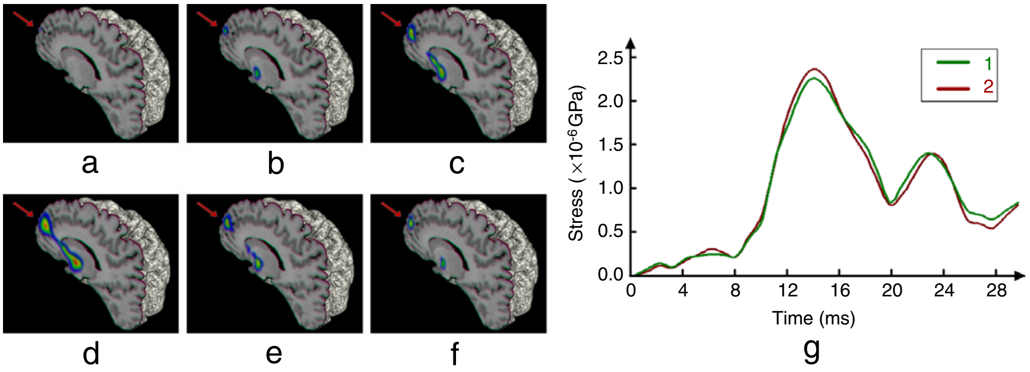

Fig. 13. (a-f) Another brain injury simulation with a time interval of 3 ms. The blunt impact occurs at the left front lobe; (g) Comparison of stress evolution of the right thalamus under the blunt impact.

Table 1
Statistics of 3D reconstruction of brain models. The fitting error is presented by root-mean-square error.

Subject	Degree	Data points	Tetrahedra	Control points	Knots	Fitting error
A	2	60 298	2500	3 871	1683	3.0375×10^{-4}
B	3	72 357	2500	12 431	2244	2.1483×10^{-4}
C	2	79 593	4320	6 525	2769	1.9743×10^{-4}
D	3	86 226	4320	21 117	3682	1.5290×10^{-4}

Table 2
Physical parameters and statistics of brain biomechanic simulations.

Application	μ (kg/m^3)	γ (1/s)	α (GPa)	β (GPa)	Δt (ms)	Time (hh:mm:ss)
Shifting simulation	1.04×10^{-6}	400	0.6570	0.2266	75	00:24:20
Impact simulation	1.04×10^{-6}	400	0.6570	0.2266	3	00:22:33

Fig. 12 shows two stress evolution curves of one landmark inside the right thalamus under the specified blunt impact in Fig. 11. The green one is the ground truth obtained from the real biomechanic experiments and the red one is the result simulated using our framework. The result curves demonstrated that our simulation can obtain an accurate and satisfactory result, which has great potential for computer-aided diagnosis of brain injury under blunt impact.

Fig. 13 illustrates another experiment regarding brain injury prediction. A different patient is undergoing a similar blunt impact on the left frontal lobe. The simulation shows that besides the spot under the direct impact, the thalamus is another area where bleeding may happen, which also coincides with the ground truth fact.

6. Experimental results

We have implemented a prototype system on a Dell Precision Workstation T7400, which has dual Xeon CPUs with Quad Cores and 4GB RAM. The system is written in VC++ and VTK 4.2. We performed experiments on several brain datasets. In order to compare the reconstruction qualities for patient-specific cases, we uniformly sample the brain geometric and physical fields into a unit cube.

Table 1 shows the configuration of DSVSS volumes reconstructed from different datasets. The performance statistics of our fitting algorithm is also included. From the table, one can observe that, compared with discrete mesh representation, our spherical volumetric simplex spline based representations have low storage requirements and can achieve high accuracy, e.g., fitting r.m.s. error $\leq \times 10^{-4}$. The high computational cost is the challenging aspect of our algorithm. However, in practice, by applying multiresolution and multi-thread implementation of the geometric elements, the time cost can be greatly reduced.

Table 2 shows the performance of a DSVSS framework applied to brain datasets. Both brain biomechanic behaviors, brain shifting and brain injury, are simulated. Note that all the units in this table are metric. In the table, only the physical properties of white matter are listed. For the physical properties of different type of brain tissues, readers are referred to [38]. The averaged, overall computational time for the entire simulations demonstrated in Fig. 10 (brain shifting) and Fig. 11 or 13 (brain injury simulation) are listed in the table. Faster overall computations can be achieved by increasing the simulation time interval. The simulation results on brain shifting and brain injury prediction have been already illustrated in Section 5. Overall, our brain simulation results exhibit a reliable approximation of how brain shifting behaves and how brains could be injured in the real world when inputting the real material parameters.

7. Conclusion

In this paper, we have developed a novel simulation framework based on dynamic spherical volumetric simplex splines. We have introduced an automatic and accurate algorithm to fit the digital models of real-world objects with a single spherical volumetric simplex spline which can represent with accuracy geometric and material properties of objects simultaneously. With the integration of Lagrangian mechanics, a dynamic volumetric simplex spline representing the real-world object can accurately simulate its physical behavior. We have applied the framework in a biomechanical simulation of the brain, such as brain shifting during surgery and brain injury under sudden impact. We have compared the simulated results with the ground truth obtained through interactive magnetic resonance imaging and the ground truth from real biomechanical experiments. The experimental results have demonstrated the excellent performance of our technique, which can be effectively used in deformation-based brain simulation and simulation-based diagnosis/assessment. The robustness and accuracy result from the tight integration of the geometric and material properties into the simulation. In the near future, we will investigate more powerful simulation schemes based on our novel digital representations. Hierarchical simulation will also be explored to speed up the simulation for real-time applications. On the application side, we will develop a DSVSS model of an entire head, which allows us to simulate more sophisticated behaviors of the brain.

Acknowledgements

The authors would like to thank Dr. Liying Zhang for providing us the data and biomechanic experimental results for evaluation. This work is supported in part by the research grants awarded to Dr. Jing Hua, including the National Science Foundation grants IIS-0713315 and CNS-0751045, the National Institute of Health grant 1R01NS058802-01A2 and 2R01NS041922-05A1, the Michigan Technology Tri-Corridor grants MTTC05-135/GR686 and MTTC05-154/GR705, and the Michigan 21st Century Jobs Funds 06-1-P1-0193.

References

[1] Hua J, He Y, Qin H. Multiresolution heterogeneous solid modeling and visualization using trivariate simplex splines. In: Proceedings of the ninth ACM symposium on solid modeling and applications. 2004. p. 47–58.

[2] Maguire G, Noz M, Rusinek H, Jaeger J, Kramer E, Sanger J, et al. Graphics applied to medical image registration. IEEE Computer Graphics Application 1991;11:20–8.

[3] de Boor C. Splines as linear combinations of B-splines. In: Approximation Theory II. New York: Academic Press; 1976. p. 1–47.

[4] Greiner G, Seidel H-P. Modeling with triangular B-splines. IEEE Computer Graphics and Applications 1994;14(2):56–60.

[5] Dahmen W, Micchelli C, Seidel H-P. Blossoming begets B-spline bases built better by B-patches. Mathematics of Computation 1992;59(199):97–115.

[6] Pfeifle R, Seidel H-P. Fast evaluation of quadratic bivariate DMS spline surfaces. In: Proceedings of graphics interface '94. 1994. p. 182–9.

[7] Pfeifle R, Seidel H-P. Scattered data approximation with triangular B-splines. In: Advance course on fairshape. 1996. p. 253–63.

[8] Pauly M, Gross M, Kobbelt L. Efficient simplification of point-sampled surfaces. In: IEEE visualization 02 proceedings. 2002. p. 163–70.

[9] Hua J, Qin H. Haptic sculpting of volumetric implicit functions. In: Proceedings of the ninth Pacific conference on computer graphics and applications (Pacific graphics). 2001. p. 254–64.

[10] Hua J, Qin H. Haptics-based dynamic implicit solid modeling. IEEE Transactions on Visualization and Computer Graphics 2003;10.

[11] Hua J, He Y, Qin H. Trivariate simplex splines for inhomogeneous solid modeling in engineering design. ASME Transactions: Journal of Computing and Information Science in Engineering 2005;5(2):149–57.

[12] Tan Y, Hua J, Dong M. 3D reconstruction from 2D images with hierarchical continuous simplicies. The Visual Computer 2007;23(9–11):905–14.

[13] Terzopoulos D, Fleischer K. Deformable models. The Visual Computer 1988; 4(6):306–31.

[14] Celniker G, Gossard D. Deformable curve and surface finite elements for free-form shape design. Computer Graphics 1991;25(4):257–66.

[15] Bloor M, Wilson M. Representing PDE surfaces in terms of B-splines. Computer-Aided Design 1990;22(6):324–31.

[16] Welch W, Witkin A. Variational surface modeling. Computer Graphics 1992; 26(2):157–66.

[17] Terzopoulos D, Qin H. Dynamic NURBS with geometric constraints for interactive sculpting. ACM Transactions on Graphics 1994;13:103–36.

[18] Qin H, Terzopoulos D. Dynamic NURBS with geometric constraints for physics-based shape design. Computer Aided Design 1995b;27(2):111–27.

[19] Qin H, Terzopoulos D. Dynamic manipulation of triangular B-splines. In: Proceedings of third symposium on solid modeling and applications 1995. p. 351–60.

[20] Zhang Y, Hughes T, Bajaj C. Automatic 3D mesh generation for a domain with multiple materials. In: Proceedings of the 16th international meshing roundtable. 2007. p. 367–86.

[21] Zhang L, Yang K, Dwarampudi R, Li KOT, Kun C, Hardy W, et al. Rencent advances in brain injury research: A new human head model development and validation. Stapp Car Crash Journal 2001;45:369–93.

[22] Gefen A, Margulies S. Are in vivo and in situ brain tissues mechanically similar. Journal of Biomechanics 2004;37:1339–52.

[23] Coats B, Margulies S. Material properties of porcine parietal cortex. Journal of Biomechanics 2006;39:2521–5.

[24] Ruan J, Khalil T, King A. Human head dynamic reponse to side impact by finite element modeling. Transactions of the ASME 1991;113:276–83.

[25] Kang H, Willinger R, Diaw B, Chinn B. Modeling of the human head under impact conditions: A parametric study. In: Proceedings of 41th stapp car crash conference. 1997.

[26] Edelsbrunner H. In: Ciarlet PG, Iserles A, Kohn RV, Wright MH, editors. Geometry and topology for mesh generation. Cambridge University Press; 2001.

[27] Wang Y, Gu X, Chan T, Thompson P, Yau S. Volumetric harmonic brain mapping. In: ISBI'04: IEEE internaltional symposium on biomedical imaging: Macro to nano. 2004. p. 1275–8.

[28] Wang Y, Gu X, Yau S. Volumetric harmonic map. Communications in Information and Systems 2004;3(3):191–202.

[29] Li X, Guo X, Wang H, He Y, Gu X, Qin H. Harmonic volumetric mapping for solid modeling applications. In: In Proceedings of the 2007 ACM symposium on solid and physical modeling. 2007. p. 109–20.

[30] Gu X, Wang Y, Chan T, Thompson P, Yau S. Genus zero surface conformal mapping and its application to brain surface mapping. Information Processing in Medical Imaging 2003;172–84.

[31] Wang Y, Gu X, Chan T, Thompson P, Yau S. Intrinsic brain surface conformal mapping using a variational method. In: SPIE international symposium on medical imaging. 2004.

[32] Gossick B. Hamilton's principle and physical systems. New York (London): Academic Press; 1967.

[33] Halstead M, Kass M, DeRose T. Efficient, fair interpolation using catmull-clark surfaces. In: Computer Graphics proceedings, Annual conference series. In: Proc. ACM Siggraph 93. 1993. p. 35–44.

[34] Terzopoulos D. Regularization of inverse visual problems involving discontinuities. IEEE Transactions on Pattern Analysis and Machine Intelligence 1986; 8:413–24.

[35] Terzopoulos D, Platt J, Barr A, Fleischer K. Elastically deformable models. Computer Graphics 1987;21(4):205–14.

[36] Moreton H, Sequin C. Functional optimization for fair surface design. Computer Graphics 1992;26(2):167–76.

[37] Kardestuncer H. Finite element handbook. New York: McGraw-Hill; 1987.

[38] Zhang L, Bae J, Hardy W, Monson K, Manley G, Goldsmith W, et al. Computational study of the contribution of the vasculature on the dynamic response of the brain. Stapp Car Crash Journal 2002;46:145–64.

[39] Ting T. Anisotropic elasticity. Oxford University Press; 1996.

[40] Press W, Flanney B, Teukolsky S, Verttering W. Numerical recipes: The art of scientific computing. cambridge: Cambridge University Press; 1986.

[41] Nimsky C, Ganslandt O, Cerny S, Hastreiter P, Greiner G, Fahlbusch R. Quantification of visualization of, and compensation for brain shift using intraoperative magnetic resonance imaging. Neurosurgery 2000;47(5):1070–80.

[42] Ferrant M, Warfield S, Nabavi A, Jolesz F, Kikinis R. Registration of 3D intraoperative MR images of the brain using a finite element biomechanical model. In: MICCAI. 2000. p. 19–28.

[43] Miga M, Sinha TK, Cash DM, Galloway R, Weil R. Cortical surface registration for image-guided neurosurgery using laser-range scanning. IEEE Transactions on Medical Imaging 2003;22(8):973–85.