Computers & Graphics 000 (2017) 1–11

Contents lists available at ScienceDirect

Computers & Graphics

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

Special Section on Cyberworlds 2016

Hessian-constrained detail-preserving 3D implicit reconstruction from raw volumetric dataset

Shuai Li[a], Dehui Yan[a], Xiangyang Li[b], Aimin Hao[a,*], Hong Qin[c]

[a] State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, China
[b] Thomson Reuters, Beijing, China
[c] Department of Computer Science, Stony Brook University, USA

ARTICLE INFO

Article history:
Received 15 October 2016
Revised 26 December 2016
Accepted 16 January 2017
Available online xxx

Keywords:
3D shape modeling
Volumetric dataset
Implicit reconstruction
Hessian constraints
Generalized radial basis functions

ABSTRACT

Massive routinely-acquired raw volumetric datasets are hard to be deeply exploited by downstream applications due to the challenges in accurate and efficient shape modeling. This paper systematically advocates an interactive 3D shape modeling framework for raw volumetric datasets by iteratively optimizing Hessian-constrained local implicit surfaces. The key idea is to incorporate contour based interactive segmentation into the generalized local implicit surface reconstruction. Our framework allows a user to flexibly define derivative constraints up to the second order via intuitively placing contours on the cross sections of volumetric images and fine-tuning the eigenvector frame of Hessian matrix. It enables detail-preserving local implicit representation while combating certain difficulties due to ambiguous image regions, low-quality irregular data, close sheets, and massive coefficients involved extra computing burden. To this end, we propose several novel technical elements, including data-specific importance sampling for adaptive spherical-cover generation, close sheet determination based on distinguishable local samples, and parallel acceleration for local least squares fitting. Moreover, we conduct extensive experiments on some volumetric images with blurry object boundaries, and make comprehensive, quantitative performance evaluation between our method and the state-of-the-art radial basis function based techniques. And we also apply our method to two practical applications. All the results demonstrate our method's advantages in the accuracy, detail-preserving, efficiency, and versatility of shape modeling.

1. Introduction and motivation

A great number of volumetric datasets have been routinely acquired every day and their qualities are varying tremendously, without proper processing they could not be directly utilized, which could prevent rich and numerous volumetric datasets from being deeply exploited by users without specialized domain/geometry/graphics knowledge. Specifically, 3D shape modeling from raw volumetric datasets plays a vital role in many downstream applications, including shape understanding, physical simulation, interactive design and editing, geometric analysis, and visualization. However, there are still tremendous difficulties spanning from efficient and accurate volume segmentation to detail-preserving 3D shape reconstruction.

For 3D shape modeling from raw volumetric dataset, volume segmentation is inevitable. So far, the related approaches can be classified into clustering based methods and interaction based methods. Although clustering based methods have advantages in efficiency and user input, which usually require a few input parameters, it is well-known that they are hard to achieve accurate segmentation results due to lacking fine-grained priors. Meanwhile, the involved parameters' selection commonly requires domain-specific expertise to distinguish the foreground region. Specifically, Owada et al. [1] pointed out that segmentation heavily depends on the user's subjective interpretation, which indicates that effective user interpretation indeed facilitates accurate segmentation.

Benefiting from user-specified contours, the interactive volumetric shape modeling methods usually involve a reconstruction procedure, typically based on certain implicit functions that are commonly considered to be robust to noise, topologically flexible to be interpolated and extrapolated, and easily to be converted to other geometry representations. In the past two decades, a number of implicit reconstruction methods have been proposed, mainly including local implicit methods and global implicit methods. Local implicit methods, such as moving least squares [2] and multi-level partition of unity implicits [3], have well-known advantages in high accurate representation and low computation cost. However, they tend to produce deformed messy shapes when handling

* Corresponding author.
E-mail addresses: ham@buaa.edu.cn (A. Hao), qin@cs.stonybrook.edu (H. Qin).

http://dx.doi.org/10.1016/j.cag.2017.01.001
0097-8493/

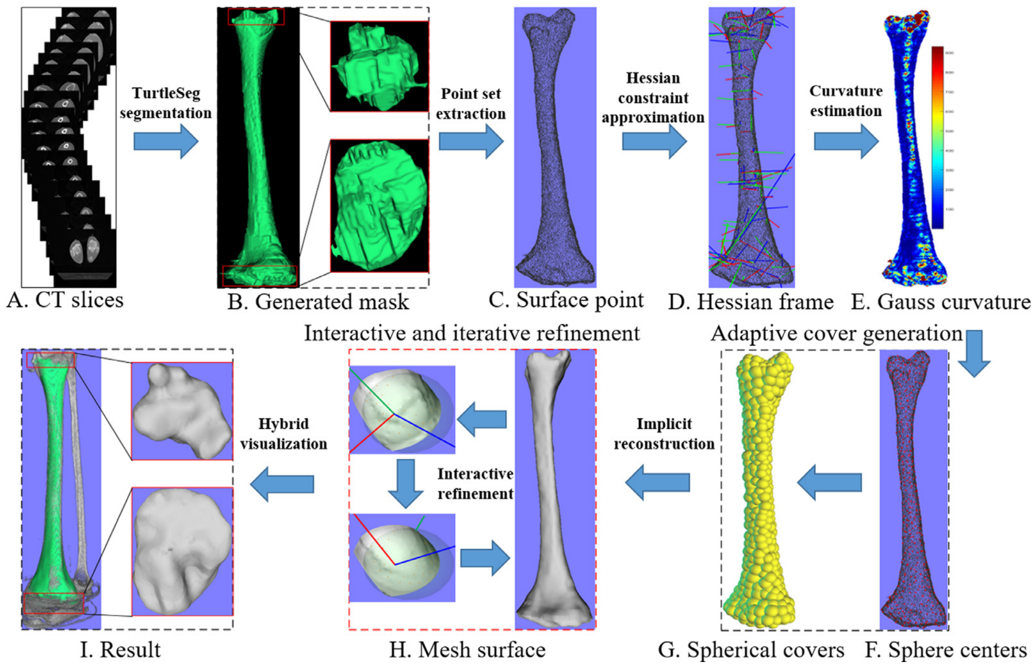

Fig. 1. Flow chart of our method. (A): Original CT slices; (B): The segmentation mask generated by TurtleSeg [24]; (C): The extracted surface points; (D): The Hessian frames; (E): Gaussian curvature; (F,G): Spherical cover generation for local least-squares implicit surface reconstruction; (H): The reconstructed mesh surface, the red box shows the iterative segmentation refinement via user-controllable Hessian-constrained implicits optimization; (I) The finally-segmented mesh surface, which is visualized via hybrid rendering over original volume data.

low-quality data with noise, outliers, and uneven sampling density. Global implicit methods, including RBF based approaches [4], variational approach [5], graph-cut approach [6], Hermite RBF Implicits (HRBF) [7], etc, are less sensitive to data quality because of their global nature, and thus can effectively alleviate the problems encountered by the local implicit methods. However, they inevitably suffer from low-accuracy reconstruction and computational burden involved in large system solving.

This paper systematically advocates a detail-preserving 3D shape modeling method from raw volumetric dataset by generalizing local HRBF based on Hessian constraints and incorporating them into iterative optimization of local implicit surface. In particular, the salient contributions of this paper can be summarized as follows:

- We propose an interactive 3D shape modeling framework based on iterative optimization of local implicit functions, which affords users an intuitive interface to edit the sample points and their accompanying curvature-related constraints over volumetric dataset, and thus gives rise to more accurate results.
- We formulate a new local least squares based RBF implicit by incorporating the second-derivative Hessian constraints, which facilitates detail-preserving implicit surface reconstruction over the jagged boundary voxels of the segmented volumes.
- We design an adaptive spherical cover generation scheme to adaptively determine the supporting domain of local implicits, which guarantees to effectively distinguish the local supporting domain of close sheets and respects the sharp features during interactive 3D shape modeling.

This paper is an extended version of the full paper [8] in 2016 International Conference on CYBERWORLDS. Compared to the conference paper, several detailed contexts and some new experiments are added. For example, we add Fig. 3 to illustrate iterative local Hessian constraint editing, and give further discussions about the method and experiment results analysis in Section 6. And Fig. 5 is given to intuitionally illustrate the global blending process. Meanwhile, following the pipeline in Fig. 1, we detail the main data structures in Tables 1 and 2 and the overall numerical implementation of our prototype system in Algorithm 1, of which, the algorithms are explained in detail step by step. Moreover, we design several intermediate and integrated experiments to fully verify method's effectiveness. At last, more details about our paper's practical applications in medicine and industry are introduced, which prove our paper's high practical utility.

Table 1
The structure of global data.

Identifie	Meaning
Point	The array of surface points
Normal	The array of normals
Hessian	The array of Hessian matrixs
ps_tree	The kd-tree of the point set
sc_tree	The kd-tree of the spherical cover center
Sphere	The array of spherical covers
editP	The index of the edited points
editS	The index of the edited spheres

Table 2
The structure of a local sphere.

Identifie	Meaning
$\mathbf{c}$	The center of the sphere
r	The radius of the sphere
Samples	The index of the samples
Parameters	The approximate parameters
w	The approximate error

2. Related work

Relevant to the central theme of this paper, we now briefly review previous works in two aspects: interactive contour assisted segmentation and implicit reconstruction.

Please cite this article as: S. Li et al., Hessian-constrained detail-preserving 3D implicit reconstruction from raw volumetric dataset, Computers & Graphics (2017), http://dx.doi.org/10.1016/j.cag.2017.01.001

Algorithm 1: Numerical Implementation of Interactive 3D Shape Modeling.

input : The segmentation mask volume.
output: The 3D shape modeling result.
1. Preprocess point set.
2. Generate adaptive spherical cover.
  Invoke Algorithm 2 with T_{err}.
3. Solve local least squares systems in parallel.
  Determine local samples.
  Invoke Algorithm 3 for all spherical covers.
4. Refine surface interactively and iteratively.
  Extract global iso-surface.
  while(not be satisfied with the result)
  {
    (1) Set controlling region.
    (2) Extract local iso-surface.
    (3) Interactively edit Hessian matrix according to Eq. (11).
    (4) Update the spheres that cover the controlling region by Algorithm 3.
    (5) Update global iso-surface.
  }

Interactive contour assisted segmentation. Considering that fully-manual segmentation process is laborious and time-consuming, while fully-automatic algorithms (include thresholding [9], k-means clustering [10], mean-shift [11]) are still open for accurate segmentation, it is imperative to further seek a balance between segmentation accuracy and the complexity of user input. The contour-based method is designed to directly specify contours in ambiguous regions of images with expert knowledge. For example, De Bruin et al. [12] proposed a method to generate boundary surfaces by directly connecting well-organized contour vertices. Aliroteh et al. [13] proposed a SketchSurface system, which allows users to draw closed contours on the parallel cross-sectional planes of volumetric images, and subsequently employed quick-hull algorithm and active contour iterations for segmentation. Liu et al. [14] initialized boundary surfaces using a Voronoi diagram based algorithm and then smoothed them. Motivated by physics-based deformable models, Kass et al. [15] proposed the active contour method, and its many extensions have also been developed [16]. Such methods can delineate object boundaries by making the boundary deform, driven by the internal energy sensitive to the boundary shape and the external energy sensitive to local image features. However, since these methods commonly depend on direct meshing and require well-organized contours, it is difficult to manage open or non-planar contours. Instead, TurtleSeg [17] is an interactive segmentation tool designed for 3D medical images, by interactively contouring on some sparse slices, and the rough segmentation result will be generated automatically. To respect more features, it needs to carefully place dense contours on the cross planes, even so, the segmented object may still appear jagged.

Implicit 3D reconstruction. The central idea of implicit reconstruction is to generate certain signed scalar field from the contour-constrained point set [18], and then extract the zero iso-surface. Due to its potentials to satisfactorily handle sparse sample points, great attention has been paid to RBF [4]. However, early scalar field generating methods generally require two sophisticated offset points to ensure the existence of a non-null interpolation implicit function. By incorporating normals into the problem formulation, Pan et al. [5] proposed Hermite variational implicit surface, and Macedo et al. [7] proposed Hermite Radial Basis Function (HRBF), which is a particular case of Hermite–Birkhoff interpolation with RBF. Wu and Wendland [19,20] introduced compactly supported RBF to arrive at a sparse linear system, but they are sensitive to the quality of input data and lack extrapolation ability across large holes. Although such global-support RBFs [4,21] are less sensitive to the quality of input data, they require to solve large and dense matrix system. Besides, Ijiri et al. [22] proposed a system to refine volumetric shape surface by piecewise fine-tuning curved contours, and they also introduced a new implicit method [23] to evaluate the scalar field in spatial-range domain. However, these methods heavily depend on the quality of user-controllable contours, which is difficult for novice users with little expert knowledge.

In summary, the existing implicit reconstruction based 3D modeling methods still lack comprehensive abilities to simultaneously handle smoothness control, sharp-detail preservation, and efficient computation. In principle, it requires a globally flexible and locally accurate way to respect sharp features by taking high-order directional derivatives, gradients, and scalar value interpolation into account simultaneously.

3. Method overview

Our 3D shape modeling framework aims to provide a way for users to edit the intermediately-segmented surface via adjusting the eigen-system of Hessian matrix. Fig. 1 shows the pipeline of our method. Given original volumetric slices (Fig. 1(A)), we use the TurtleSeg method [24] to generate a rough segmentation mask (Fig. 1(B)), which can help users focus on the definition of second-derivative constraints around sharp-feature areas.

Since the mask is just a rough segmentation result, as indicated by the red box in Fig. 2(A), it inevitably contains errors (Fig. 2(B)). To edit the point set via intuitive interaction, we resort to placing contours on the cross-sections (Figs. 2(C)–(F)). As shown in Fig. 2, this process is intuitive and does not require much expert knowledge.

Fig. 1(D) demonstrates the frames of Hessian matrix at some points, and we conduct local quadric least squares fitting to approximate the Hessian matrix (see Section 4.2 for details). We first approximate the implicit reconstruction locally, and then blend such local implicits together. We propose an adaptive spherical cover generation method to divide the original volume domain into overlapping spherical regions based on the importance sampling of Gauss curvature (Fig. 1(F) and (G)).

As the spherical covers are independent of each other, we design a parallel algorithm to solve the Hessian-constrained least squares RBF systems in a spherical-cover-wise way. Figs. 1(H) and 3 illustrate our interactive and iterative manipulation procedures, wherein we can locally edit the Hessian constraints and update the scalar field until we are satisfied with the results (Fig. 1(I)).

4. Volumetric data preprocessing

Given a roughly-segmented result, we conduct a series of preprocessing to convert the volume mask into local supporting domains, including potential surface point set extraction, initialization of Hessian constraints, adaptive spherical cover generation, and determination of local supporting point samples, which are detailed as follows.

4.1. Extraction of potential surface point set

To improve shape modeling accuracy and simplify the complexity of user input, we use a binary volumetric mask as input and convert it into a potential surface point set. As the object region is labeled with 1 and others with 0 in the binary mask, we extract the voxel, which is labeled with 1 and has at least one of its 26

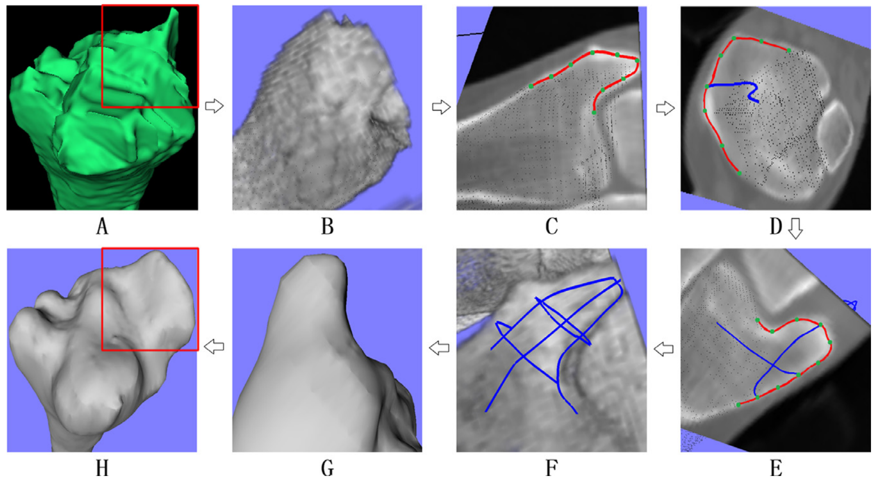

Fig. 2. Illustration of the interactive constraint manipulation. (A): The rough segmentation mask with obvious errors; (B): The potential surface points extracted from the segmentation mask; (C,D,E,F): Interactively placing contours on cross sections; (G,H): The final shape modeling results.

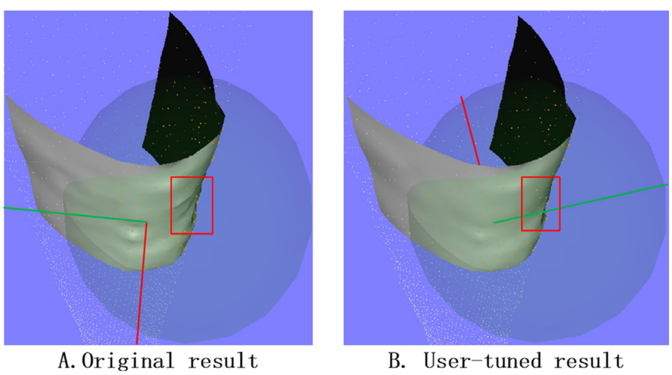

Fig. 3. Illustration of influences caused by iterative local Hessian constraint editing.

neighbors labeled with 0, as potential surface point location. However, the obtained surface points, which depend on the volume resolution, may be redundant for surface reconstruction, so we adopt Poisson disk sampling to perform down-sampling. Considering the boundaries may be jagged, we further adopt the Weighted Locally Optimal Projection operator (WLOP) method [25] to reduce noise and outliers of the potential surface point set. At last, we employ the method proposed by Liu et al. [26] to calculate the normals of the point set. Finally, we can convert the input mask into a point set with normals, which will serve as the input of our subsequent processing.

4.2. Initialization of Hessian constraints

As we wish to incorporate the Hessian information into our segmentation, we need to approximate the Hessian matrix from the point set. To facilitate the subsequent interactive manipulation over the scattered surface points, we initialize the Hessian constraints by locally fitting a quadric surface $G_i(\mathbf{x})$ at each sample point $\mathbf{x}_i$.

$$G_i(\mathbf{x}) = \sum_{j=1}^{10} \gamma_j * p_j,\ p_j \in \{1, x, y, z, \dots, x^2, y^2, z^2\}, \tag{1}$$

$$\arg\min_{\mathbf{q}} \sum_{\mathbf{x}_k \in nei(\mathbf{x}_i)} (G_i(\mathbf{x}_k)^2 + \| \nabla G_i(\mathbf{x}_k) - \mathbf{n}_k \|^2). \tag{2}$$

Here $\mathbf{q} = (\gamma_1, \dots, \gamma_{10})$ is the coefficient to be determined, p_j is the quadric polynomial basis, $nei(\mathbf{x}_i)$ represents the neighbor of $\mathbf{x}_i$ and $\mathbf{n}_i$ is the normal associated with point $\mathbf{x}_i$. In general, the number of neighbors is set to be around 20. Therefore, we can get the Hessian matrix of point $\mathbf{x}_i$ as follows:

$$\mathbf{H}_i = \begin{pmatrix} 2\gamma_8 & \gamma_5 & \gamma_6 \\ \gamma_5 & 2\gamma_9 & \gamma_7 \\ \gamma_6 & \gamma_7 & 2\gamma_{10} \end{pmatrix}, \tag{3}$$

where $2\gamma_8 = \frac{\partial^2 G_i}{\partial x^2}$, $2\gamma_9 = \frac{\partial^2 G_i}{\partial y^2}$, $2\gamma_{10} = \frac{\partial^2 G_i}{\partial z^2}$, $\gamma_5 = \frac{\partial^2 G_i}{\partial x \partial y}$, $\gamma_6 = \frac{\partial^2 G_i}{\partial x \partial z}$, $\gamma_7 = \frac{\partial^2 G_i}{\partial y \partial z}$.

4.3. Adaptive spherical cover generation

Since the global RBF methods commonly suffer from computation overhead and tiny details missing, we propose a generalized local least squares RBF by integrating Hessian constraints. The local supporting domains are represented as overlapping spherical covers and each spherical cover has five components: the center and radius are defined adaptively, the supporting samples are determined upon the normal direction of the center (see Section 4.4 for details), the parameters and weights are determined by Hessian-constrained local least squares RBF implicits (see Section 5.1 for details).

In order to respect local sharp features, we design an adaptive spherical cover generation method by making points with high-curvature stand out to serve as the spherical cover centers. For each spherical cover with center $\mathbf{c}$, we determine its radius r as follows.

$$Q(\mathbf{c}, r, \mathbf{x}) = \sum_j w(\| \mathbf{p}_j - \mathbf{c} \| / r)(\mathbf{n}_j \cdot (\mathbf{x} - \mathbf{p}_j))^2. \tag{4}$$

Here it computes a sum of the squared distances from point $\mathbf{x}$ to the tangent planes at the sample point $\mathbf{p}_j$ within the spherical region $\| \mathbf{p}_j - \mathbf{c} \| \le r$. $w(x)$ is a weight function defined as $w(x) = (1-x)_+^4(4x+1)$. If r in Eq. (4) is fixed, the minimum of $Q(\mathbf{c},\ r,\ \mathbf{x})$ can be easily found by solving a linear system, with $\mathbf{x}_{min} = \mathbf{x}_{min}(r)$. And the error function is defined as $E(r) = \frac{1}{L}\sqrt{Q(\mathbf{c}, r, \mathbf{x}_{min})}$, which measures how curved the reconstructed surface is inside the sphere $\| \mathbf{x} - \mathbf{c} \| \le r$. And L is the main diagonal length of the bounding box of point set $\mathcal{P}$. Since we expect to make r as large as possible while maintaining certain accuracy, we determine r by solving the equation with a specified accuracy T_{err} as

$$E(r) = T_{err}. \tag{5}$$

Please cite this article as: S. Li et al., Hessian-constrained detail-preserving 3D implicit reconstruction from raw volumetric dataset, Computers & Graphics (2017), http://dx.doi.org/10.1016/j.cag.2017.01.001

Here the only thing we need to pay attention to is the parameter T_{err}, which influences the number of spherical covers and the number of samples within each sphere. Algorithm 2 documents the details of our adaptive spherical cover generation procedure.

Algorithm 2: Adaptive spherical cover generation.

input : Original point set $\mathcal{P}$.
output: Spherical covers with center and radius.
1. Set all points in $\mathcal{P}$ as uncovered ones and initialize T_{err}.
2. Select the point $\mathbf{c}_i$ with maximal Gaussian curvature from uncovered point set.
3. For selected point $\mathbf{c}_i$, determine the radius r_i of sphere s_i centered at $\mathbf{c}_i$ according to Eq. 5.
4. Determine the sample points $\mathcal{S}_i$ that belong to the sphere s_i (see Section 4.4 for details).
5. Label the points of $\mathcal{S}_i$ as covered ones.
6. Terminate the process until there are no more uncovered points, otherwise go back to **2**.

Algorithm 3: Parallel Solver of the Local Implicits.

input : The spherical covers.
output: The spherical covers with corresponding parameters.
OpenMP parallel for spherical covers
{
1. Initialize the parameters in Eq. 8 as $\lambda_1 = 0.9$, $\lambda_2 = 0.09$, $\lambda_3 = 0.01$.
2. Calculate matrix **A** and vector **y** according to Eq. 13.
3. Use the Eigen library˜[28] to solve the least squares system $\mathbf{Ax} = \mathbf{y}$.
}

4.4. Determination of local supporting point samples

Due to the possible existence of close sheets, the accurate local sample determination is important to avoid self-intersecting artifacts. As shown in Fig. 4(A), the two curves represent the cross section of an object, wherein the red sphere s_i means the local fitting region with center O and radius r. In order to correctly select the samples belonging to s_i, we first assign the points within the sphere into small cells, wherein each cell only contains a few points with similar normal directions (the cell containing one sample at most). And then we convert these cells into an adjacency graph. Here the adjacency is defined as the one-ring neighbor with similar normal orientation. Taking Fig. 4(A) as an example, the one-ring neighbors of $C1$ are $C2$, $C3$, $C5$, $C6$, $C7$, but only $C2$ and $C3$ have the similar normal orientations with $C1$, so we consider $(C1, C2)$ and $(C1, C3)$ are adjacent. After that, we can get a geodesic-like distance of (Ci, Cj). Next, we consider sphere s_i, denoting its sample point set as $\mathcal{S}_i$, we first add its center into $\mathcal{S}_i$, and then add the point within the cell $C1$ into $\mathcal{S}_i$, of which, such point is covered by s_i and has the similar orientation with the point in $\mathcal{S}_i$. Consider its adjacency cells $C2$, $C3$ in the same way, and repeat this step until no more points covered by sphere s_i can be added. Finally, we can obtain the points from $\mathcal{S}_i$ to approximate region s_i. And Fig. 4(B) illustrates this process in a toy example.

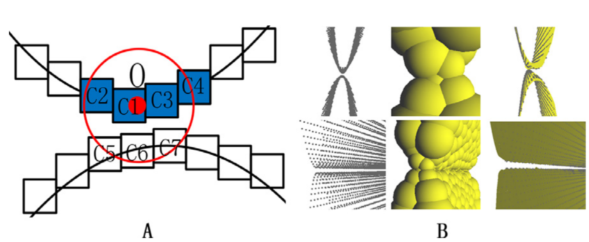

Fig. 4. Illustration of separable close sheets in a toy example.

5. Interactive and iterative implicit reconstruction

The construction of an implicit surface is equivalent to segmenting the domain into two regions, an interior region and an exterior region with the boundary approximating the surface point set. By incorporating Hessian matrix into the generated local least squares RBF implicits, our method provides users with an intuitive interface to refine the reconstruction result via editing the Hessian constraints interactively and updating the scalar field iteratively.

5.1. Hessian-constrained local least squares RBF implicits

Within each spherical cover s, we want to locally construct a signed-distance function $f(\mathbf{x})$ by approximating the samples of s. Thanks to the superiority of RBF in handling sparse point clouds, we employ it as our basis function to fit the implicit surface. In general, a RBF method has the following form:

$$f(\mathbf{x}) = P(\mathbf{x}) + \sum_{i=1}^{N} \lambda_i \phi(\| \mathbf{x} - \mathbf{x}_i \|), \tag{6}$$

where $P(\mathbf{x})$ is a low-order polynomial and the basic function ϕ is a real-valued function on $[0, +\infty)$, which is usually unbounded and has non-compact support. The common choice of the basic function ϕ includes the thin-plate spline $\phi(r) = r^2 log(r)$ (usually used for fitting smooth functions of two variables), the Gaussian $\phi(r) = exp(-cr^2)$ (mainly for neural network), and the multiquadric $\phi(r) = \sqrt{r^2 + c^2}$ (particular suitable for topographical data). In this paper, we choose $\phi(r) = r^3$ as a basic function, because it can better deal with three variables, with $P(\mathbf{x})$ as a linear polynomial.

Hessian matrix is a square matrix comprising the second-order partial derivatives of a scalar-valued function or a scalar field. It essentially depicts the local curvatures of a multi-variate function. We expect to preserve more sharp features and obtain smoother surface by incorporating Hessian constraints into RBF implicit reconstruction. Therefore, in contrast to solely normal information involved implicit reconstruction, within each sphere s_l, our reconstruction problem is converted into an optimization problem as follows,

$$f_l(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i \phi(\| \mathbf{x} - \mathbf{x}_i \|) + \sum_{j=1}^{4} \beta_j p_j(\mathbf{x}), \tag{7}$$

$$\arg\min_{\mathbf{q}} \sum_{k=1}^{n} \lambda_1 f_l(\mathbf{x}_k)^2 + \lambda_2 \| \nabla f_l(\mathbf{x}_k) - \mathbf{n}_k \|_2^2 + \lambda_3 \| Hf_l(\mathbf{x}_k) - \mathbf{H}_k \|_F^2, \tag{8}$$

where $p_j(\mathbf{x})$ is the linear polynomials, $\mathbf{q} = (\alpha_1, \ldots, \alpha_m, \beta_1, \ldots, \beta_4)$ is the parameter to be determined, variable m represents the number of local RBF centers, n is the number of samples, $\mathbf{n}_k$ and $\mathbf{H}_k$, respectively, represent the normal and Hessian matrix associated with the sample $\mathbf{x}_k$. And $Hf_l(\mathbf{x}_k)$ is Hessian matrix of f_l, a 3×3 matrix of the second-order partial derivatives of $f_l(\mathbf{x})$ at point $\mathbf{x}_k$, whose norm is defined as the Frobenius norm (i.e., the sum of the squares of nine entries of the matrix). The weighting parameters λ_1, λ_2, λ_3 (satisfying $\lambda_1 + \lambda_2 + \lambda_3 = 1$) control the importance of different items. The system can be solved by a least squares solver.

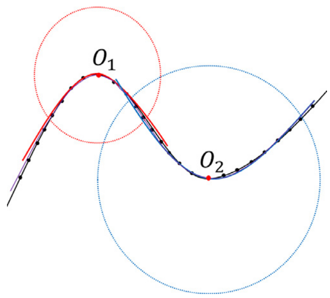

Fig. 5. The illustration of global blending process.

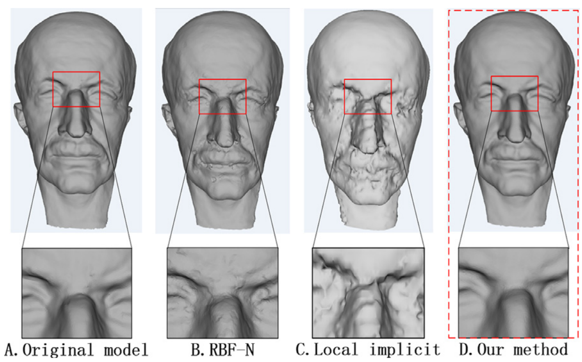

Fig. 6. The comparison of implicit shape modeling results between our method and RBF-N method over Max-Planck model. (A) shows the original model, (B) shows the RBF-N result, (C) shows the intermediate result of our method before global blending, and (D) shows our final result.

5.2. Global blending of local implicits

Since the Partition of Unity (POU) approach is typically used to integrate locally defined approximations into a global one, wherein important properties, such as the maximum error and convergence order, can be inherited from the local approximations. Therefore, for each sphere s_l, we define a parameter to evaluate the approximation accuracy of f_l within s_l:

$$err_l = \frac{\sum_{\mathbf{x}_i \in S_l} abs(f_l(\mathbf{x}_i))}{N(\mathcal{S}_l)}, \tag{9}$$

where $N(\mathcal{S}_l)$ is the number of samples in $\mathcal{S}_l$. The weight w_l of s_l is defined as $\frac{err_{max}}{err_l}$, err_{max} is the maximum of err_l. Similar to the POU method, we define our global fitting function $F(\mathbf{x})$ as follows,

$$F(\mathbf{x}) = \frac{\sum_{i \in cover(\mathbf{x})} w_i * f_i(\mathbf{x})}{\sum_{i \in cover(\mathbf{x})} w_i}. \tag{10}$$

Here $cover(\mathbf{x})$ represents the index of spheres s_i, satisfying $\| \mathbf{x} - \mathbf{c}_i \| \le r_i$.

As shown in Fig. 5, the black points represent the data of point set, O_1 and O_2 represent two local spherical covers, the red curve fits the points in O_1, and the blue curve fits the points in O_2, while the purple curve blends the two regions together. In the non-overlapping region, the blending result is almost the same with single fitting result. In the overlapping region, the blending result is in between the two single fitting results.

We verify the accuracy of our Hessian constraints involved local least squares RBF by comparing our method with only normal constraint involved least squares RBF method (RBF-N for short), which is similar to ours without Hessian constraints in Eq. (8), over some standard 3D models with ground truth. Fig. 6 shows the implicit reconstruction results of a Max-Planck model, our method produces better results than RBF-N around sharp feature areas, such as the corners of eyes and mouth. The comparison of Fig. 6(C) and (D) proves the effectiveness of our global blending procedure. Fig. 7 shows the implicit reconstruction result of a lion model, according to the enlarged figures, we can see that our method can produce smooth surface while keeping sharp features.

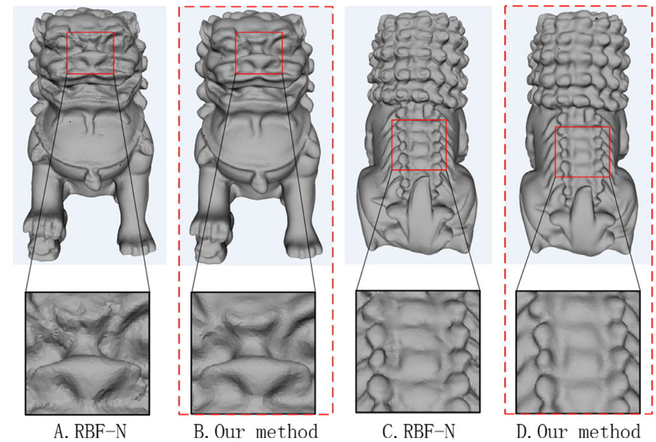

Fig. 7. The comparison of implicit shape modeling results between our method and RBF-N method over a Lion model. The first, third columns show the results of RBF-N method, and the second, fourth columns show our results.

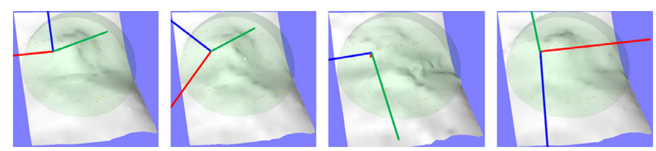

Fig. 8. The shape modeling results under different Hessian constraints and the light green color indicates the controlling sphere.

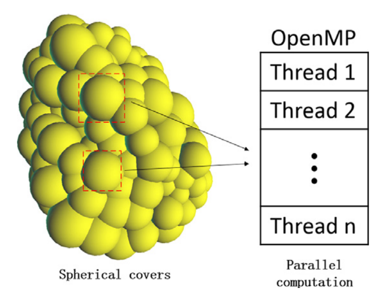

Fig. 9. Demonstration of using OpenMP to compute local implicits in parallel.

5.3. Interactive and iterative refinement

When getting an initial surface of the segmented object, we conduct refinement locally by interactively fine-tuning the frame of Hessian matrix, which corresponds to the eigenvectors of Hessian matrix. For Hessian matrix $\mathbf{H}_i$ of point $\mathbf{x}_i$, we conduct eigenvalue decomposition as $\mathbf{H}_i = \mathbf{P}\boldsymbol{\Lambda}\mathbf{P}^{-1}$, where $\mathbf{P} = (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$, $\boldsymbol{\Lambda} = diag(v_1, v_2, v_3)$ and $\mathbf{e}_j$ is the eigenvector corresponding to eigenvalue v_j. After some necessary manual tuning, we get the new frame $\{\mathbf{e}_1^{'}, \mathbf{e}_2^{'}, \mathbf{e}_3^{'}\}$, and then update Hessian matrix as

$$\mathbf{H}_i = \mathbf{P}^{'}\boldsymbol{\Lambda}\mathbf{P}^{'-1}, \tag{11}$$

where $\mathbf{P}^{'} = (\mathbf{e}_1^{'}, \mathbf{e}_2^{'}, \mathbf{e}_3^{'})$. After that, we update the local domains, re-calculate the scalar field and the iso-surface. We can iteratively repeat the refinement process until we are satisfied with the results. In fact, different manually-tuned Hessian constraints will lead to different reconstructed surface results. Of which, some may be rough, and some may be smooth. Fig. 8 shows the reconstruction results under different Hessian-constraints obtained from interactive manipulation. We can see that, the fourth result is smoother than the first three ones. However, its hard to compute the quantitative relationship between the Hessian constraints and the reconstructed surfaces. Thus, we design the interactive method for the user to finely tune the frame of Hessian matrix to get satisfying result. Actually, its a process that, the users iteratively specify the constraint according to their individual understanding on the input volumetric dataset.

5.4. Numerical implementation

Following the pipeline in Fig. 1, we detail the main data structures in Tables 1 and 2. And the overall numerical implementation of our prototype system is documented in Algorithm 1.

Point set preprocessing. The preprocessing involves some algorithms, including Poisson Disk Sampling, WLOP and normal estimation. The Poisson Disk Sampling is a simple down-sampling method. We implement the WLOP method based on the Computational Geometry Algorithms Library, and directly use the normal estimation source code proposed by Liu et al. [26]. Besides, we also provide an intuitive user interface to edit the point set, as shown in Fig. 2.

Fig. 2(A) shows the input, comparing with the volume data, we can see the significant difference within the red box. First, we set the control region by placing the center and sizing the radius of the control sphere properly. Then, we place 3D cross-sections by drawing a cut stroke on the screen and contouring on it, as shown in the Fig. 2(C–F). The red color means the editing process, and the determined curves are in green. Finally, we use the points sampled from the contours instead of the original points within the control sphere.

Adaptive cover generation. The pipeline of the adaptive spherical cover generation has been detailed in Algorithm 2. Here the only thing we need to pay attention to is the parameter T_{err}, which can influence the number of spherical covers and the number of samples within each sphere. We implement the method based on the source code provided by Ohtake et al. [27], and set $T_{err} = 0.0001$.

Local sampling. As the cover generation can not be implemented in parallel, we do local sampling right after a cover is generated. In order to achieve parallel acceleration, we store the index of each sample instead of the sample itself. Based on *ps_tree*, we can efficiently find the points within the sphere s_i with center $\mathbf{c}_i$ and take these points as candidates.

Local least-square implicits solving. To handle the time-consuming computation of Hessian constraints involved local least squares RBF implicits, we implement our method in a parallel way. To achieve this goal, we carefully design the structure of the spherical cover by storing the samples of each sphere individually. According to Eqs. (7) and (8), the optimization problem can be converted into the following form:

$$\mathbf{A}\mathbf{x} = \mathbf{y}. \tag{12}$$

For a specific sample $\mathbf{x}_i$ of sphere s_l, $\mathbf{A}$ and $\mathbf{y}$ are defined as

$$\boldsymbol{A} = \begin{bmatrix} \vdots \\ \sqrt{\lambda_1} f_l(\mathbf{x}_i) \\ \sqrt{\lambda_2}\nabla f_l(\mathbf{x}_i) \\ \sqrt{\lambda_3} H f_l(\mathbf{x}_i) \\ \vdots \end{bmatrix}, \mathbf{y} = \begin{bmatrix} \vdots \\ 0 \\ \sqrt{\lambda_2}\mathbf{n}_i \\ \sqrt{\lambda_3}\mathbf{H}_i \\ \vdots \end{bmatrix}. \tag{13}$$

As the spherical covers are independent of each other, we adopt a parallel strategy to solve the Hessian constrained least square RBF system corresponding to each sphere. We respectively set the parameters in Eq. (8) as $\lambda_1 = 0.9, \lambda_2 = 0.09, \lambda_3 = 0.01$, use the Eigen library [28] to solve the local systems, and employ Open MP to conduct parallelizing among spheres.

Iso-surface extracting. In order to extract zero iso-surface, we need to calculate a scalar filed of a specific region with a given resolution. When extracting the global mesh, we set the region to be the bounding box of the point set, and the resolution is set to be 64*64*64. For a local mesh, the region is set to be the bounding box of the controlling sphere and the resolution is set to be 16*16*16. With the region and resolution, we use Algorithm 4 to calculate the scalar field, from which we can extract the zero iso-surface using CMS method [29].

Algorithm 4: Scalar Value Calculation.

input : A point $\mathbf{p}$.
output: The scalar value of $\mathbf{p}$.
1. Find sphere index array **list**.$\| \mathbf{c}_i - \mathbf{p} \| \le r_i, i \in$ **list**;
  If not found, set **list** $= \{ind\}$ with $min_{ind}(\| \mathbf{c}_{ind} - \mathbf{p} \|)$.
2. Set temporary variables $tot_v \leftarrow 0$, $tot_w \leftarrow 0$.
3. For index k in array **list** do
  tot_v += $w_k * f_k(\mathbf{p})$.
  $tot_w + = w_k$.
  end.
4. Return the value $\frac{tot_v}{tot_w}$.

Interactive editing. First, we set the controlling sphere. Then, we pick the point within the controlling sphere and tune its frame. Finally, we update the Hessian matrix of points, re-calculate the parameters of corresponding spherical covers, and update the zero iso-surface. Though we only rotate the frame of the picked point as shown in Fig. 3, all the points within the control sphere are influenced in the same way and stored in the variable *editP*. These points, covered by controlling sphere and considered as edited points, can be found efficiently via *ps_tree* and the indexes of spheres. And *editS* can be found via *sc_tree*. As these spheres are also independent of each other, we update their parameters in a parallel way as that described in local least-square implicits solving.

6. Experimental results and evaluation

We have implemented a prototype system using C++, and all the experiments are run on a desktop with Intel Core(TM) i7-3770 CPU (3.4GHz) and 16G RAM. Table 3 lists the experimental data statistics in details. The resolution of the volumetric data determines the number of boundary voxels, i.e., original surface points, thus, our sequent processing of point set is essential. The time costs of Hessian constraints calculation, adaptive spherical cover generation and the solving of local least squares systems show the efficiency of our framework. Specifically, our adaptive spherical cover generation procedure requires to specify a local approximation tolerance T_{err}, which controls the sphere radius. Generally speaking, the larger T_{err} is, the more samples we could expect to assign to the sphere. Since we need to solve a least squares system for each sphere, the number of samples dictates the time performance of our method. However, benefitting from our OpenMP based parallel-computing strategy, it only needs to spend a few seconds to solve such local least squares RBF systems.

We demonstrate the operation and show the intermediate result of each step in the video. Please refer to the supplementary video.

Our system involves two interactive processing phases. The first phase aims to edit the extracted potential surface point set, and the second one aims to refine the surface mesh after the initial

Table 3
The experimental data statistics, including volume resolution, the number of boundary voxels, the number of surface point samples, the time cost to calculate Hessian constraints (in seconds), the time cost to generate spherical cover (in seconds), the number of the total covers of tibia, kidney, and stone models with $T_{err} = 0.0001$ and femur model with $T_{err} = 0.00007$, the average samples of each spherical cover, and the respective time cost to solve the Hessian-constrained least squares (LS) systems in parallel and serial ways (in seconds).

Model	Volume resolution	Boundary voxels	Samples	Hessian calculation	Cover generation	Total covers	Average samples	LS solver	
								Parallel	Serial
Tibia	512*512*349	30753	11507	0.613	0.121	271	85.7	4.87	15.1
Femur	512*512*220	31098	14118	0.841	0.14	708	38.7	0.45	1.49
Kidney	512*512*323	24452	11315	0.661	0.139	295	76.8	3.54	12.2
Stone	1024*1024*332	142675	14639	0.868	0.179	486	61.7	2.86	10.2

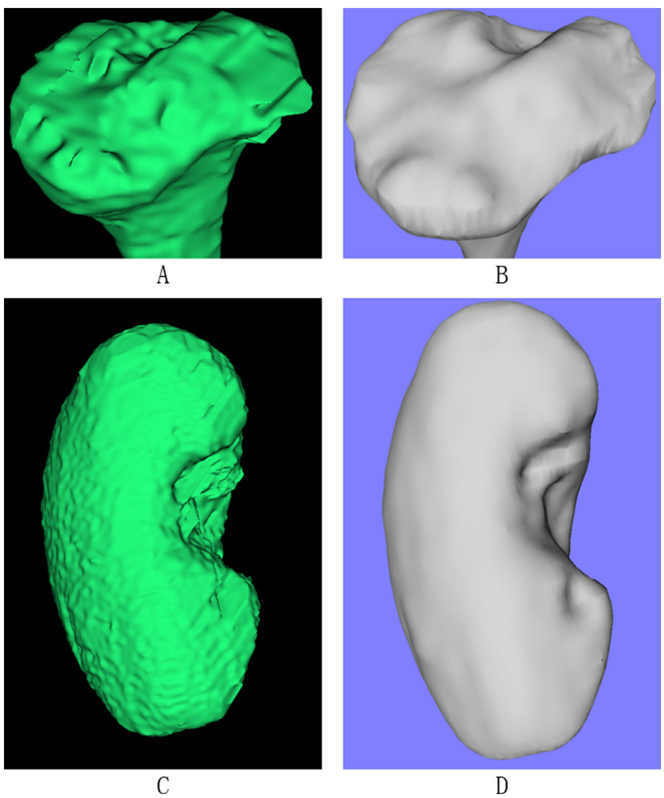

Fig. 10. The comparisons between the input masks and our final reconstruction results.

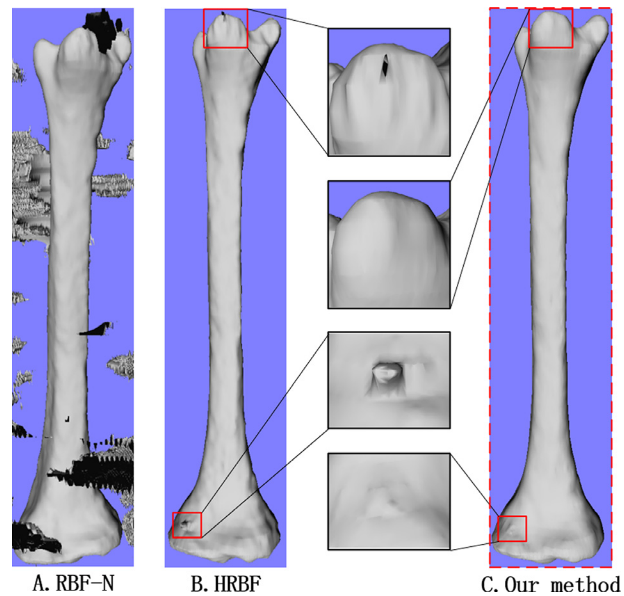

Fig. 11. The reconstructed surface comparison among RBF-N based method, HRBF based method, and our method over the tibia volume.

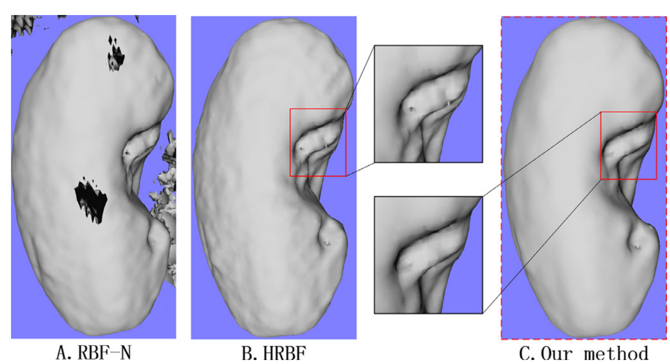

Fig. 12. The reconstructed surface comparison among RBF-N based method, HRBF based method, and our method over the kidney volume.

scalar field is obtained. Fig. 2 demonstrates an example of the first phase. Fig. 3 illustrates the second phase, of which, the red and green lines represent the eigenvectors of Hessian matrix of the selected point. Since the third eigenvalue is so small compared to the other two, the corresponding blue line is almost invisible. Fig. 3(A) shows the intermediate segmentation result, and Fig. 3(B) is the segmentation result after simply editing the constraints, wherein the local regions marked with red box are obviously smoother than Fig. 3(A). Fig. 8 compares the input segmentation mask with the final implicit reconstruction result, wherein more features are preserved benefiting from our simple but effective constraints controlling. Both Figs. 3 and 8 notably illustrate the desired surface change via specific Hessian constraints, which verifies the effectiveness of our local surface reconstruction based interactive 3D shape modeling framework.

Fig. 10 shows the comparisons between the input masks and our final reconstruction results over a bone volume and a kidney volume. Both the results indicate that our system can effectively deal with the jagged boundary and produce smooth surface with detail features.

Figs. 11 and 12 compare the shape modeling results among RBF-N based method, HRBF based method, and our method. In nature, the HRBF based method is an interpolating method, which is sensitive to noise when handling samples with outliers (Fig. 13). From Figs. 11(B) and 12(B), we can see the reconstructed surfaces from the HRBF based method exhibit many obvious artifacts. Meanwhile, Figs. 11(A) and 12(A) show that the reconstructed surfaces from the RBF-N based method also contain some artifacts caused by the inaccurate scalar field. In sharp contrast, our method can produce a more smooth shape surface. Meanwhile, Table 4 documents the statistics of error approximation on different models. Those from the RBF-N method are produced with

Please cite this article as: S. Li et al., Hessian-constrained detail-preserving 3D implicit reconstruction from raw volumetric dataset, Computers & Graphics (2017), http://dx.doi.org/10.1016/j.cag.2017.01.001

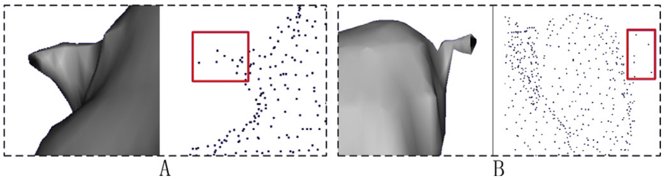

Fig. 13. The artifacts of Fig. 11(B) are shown from another view point, and the point set visualization of the corresponding regions.

Table 4
The statistics of error approximation(mm) on sample points.

Model	Method	Max	Min	Average
Kidney	HRBF	4.1e−6	−1.8e−5	1.4e−7
	RBF-N	3.8e−3	−3.6e−3	2.8e−4
	Ours	2.9e−3	−3.8e−3	4.1e−4
Tibia	HRBF	1.2e−4	−8.5e−5	5.2e−7
	RBF-N	7.8e−3	−5.8e−3	3.4e−4
	Ours	6.2e−3	−3.9e−3	5.8e−4
Max-Planck	RBF-N	3.9e−3	−5.4e−3	4.3e−4
	Ours	**3.2e−3**	**−2.4e−3**	**4.2e−4**
Lion	RBF-N	7.8e−3	−6.2e−3	2.2e−4
	Ours	**2.8e−3**	**−1.3e−3**	**1.1e−4**

$\lambda_1 = 0.9, \lambda_2 = 0.1$. Though our method and RBF-N method can obtain similar maximal, minimal, and average values, our method can generate better scalar field than the RBF-N method and guarantee to generate detail-preserving shape surfaces. When comparing with the HRBF method, our method can better deal with unexpected outliers.

Figs. 14 and 15 compare the shape modeling results among the B-HRBF based method, TurtleSeg method and our method, wherein the compared two methods are both based on interactive contours. As the B-HRBF based method allows placing contours on arbitrary cross sections and evaluates the scalar field in spatial-range domain, it is more efficient than TrutleSeg, which only involves a few contours, and whose results heavily rely on the number and quality of the input contours. In contrast, our method uses a rough segmentation result as input, which alleviates the heavy dependence on contours, thus we can focus on the interactive editing of Hessian constraints and the iterative refinement of the segmented results. According to the results shown in the cross-section slices, although such three methods can produce similar results, benefitting from integration of the second-order derivative, our method achieves more accurate shape modeling results, which preserves more details. Therefore, all the aforementioned experiments have

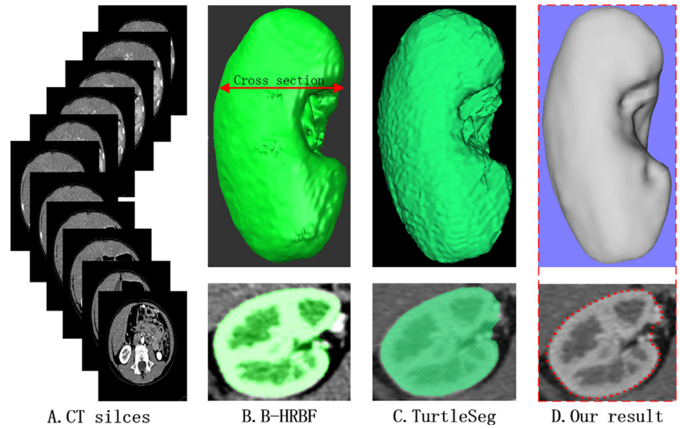

Fig. 14. Comparison among the B-HRBF method, TurtleSeg method, and ours over the kidney volume. (A) shows the input CT slice data; (B) shows the result of B-HRBF method; (C) shows the result of TurtleSeg; (D) shows our result. The second row shows the details on the cross-section slices.

demonstrated our method's advantages in accuracy, detail preservation, and high-order smoothness in a visually convincing way.

Besides, Fig. 16 and Fig. 17, respectively, show the multi-target shape modeling results of our method over a roadbed volume and a digital human volume, wherein the colored objects in the sub-figures (C) represent the surface meshes of the reconstructed objects while other objects are visualized directly using the ray-casting based volume rendering method. We use Dual marching cube method to extract the iso-surface of the colored objects. It clearly shows our method's versatility in practical applications.

Now we are cooperating with the Peking Union Medical College Hospital and the Ministry of Communications Highway Research Institute. Both of them have applied our method over their field-specific raw volumetric datasets for practical applications. As shown in Fig. 17(A), computed tomography (CT) has been being widely used in medicine to help doctors diagnose. Though visualized CT volume can help doctors a lot, it is still limited in geometric detail observing, which more or less wastes such rich data to some extent. Our method facilitates to sufficiently use these rich data for the research on individualized operation, which can well satisfy the requirements of practical medical use both in accuracy and subsequent ROI (region of interest) detailed application. As shown in Fig. 15, our method almost produces the same result with the ground truth produced by an expert doctor. Compared with the other methods, our method obviously performs the best

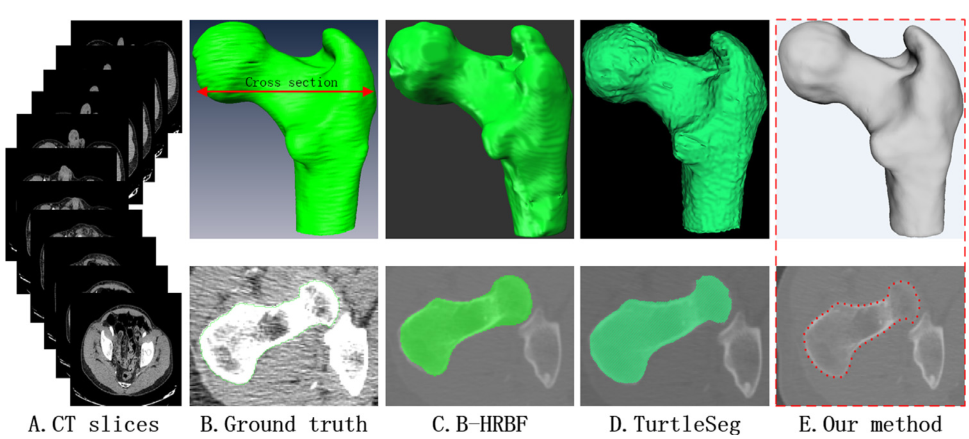

Fig. 15. Comparison among the ground truth, the B-HRBF method, TurtleSeg method and ours over the femur volume. (A) shows the input CT slice data; (B) shows the ground truth, which is manually modeled by an expert; (C) shows the result of B-HRBF method; (D) shows the result of TurtleSeg; (E) shows our result. The second row shows the details on the cross-section slices.

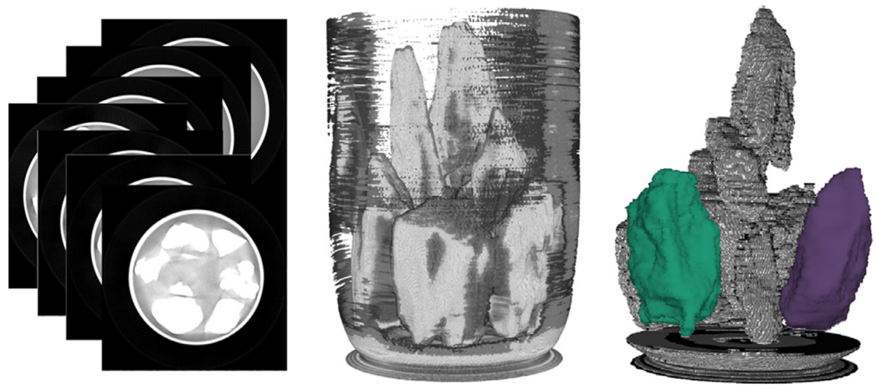

Fig. 16. The multi-target shape modeling results of our method over a roadbed volume. (A) Original CT slices; (B) Direct volume rendering; (C) The multi-target shape modeling results.

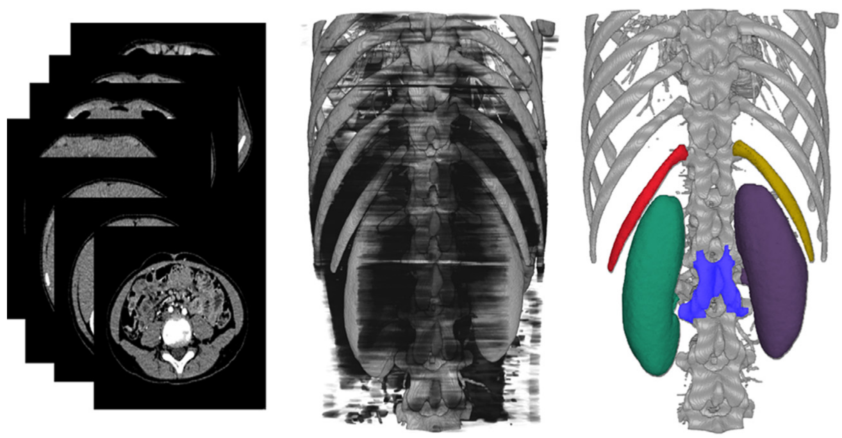

Fig. 17. The multi-target shape modeling results of our method over a digital human volume. (A) Original CT slices; (B) Direct volume rendering; (C) The multi-target shape modeling results.

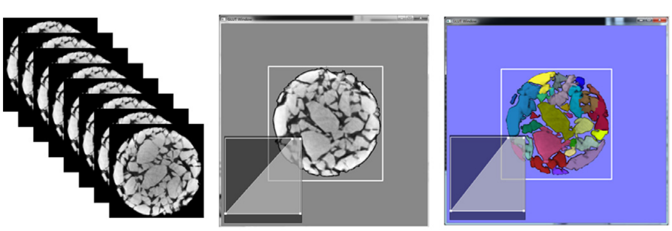

Fig. 18. The multi-target shape modeling results of our method over a more complex roadbed volume. (A) Original CT slices; (B) Direct volume rendering; (C) The multi-target shape modeling results.

in smoothness and tiny detail preserving, see Fig. 17(C). Besides, CT is also widely used in industry. Fig. 18(A) is a roadbed CT dataset. As we all know, roadbed is consisted of gravel and pitch. The distribution of gravel in the roadbed and the force analysis of the hole roadbed are important for road quality assessment. The current commercial industry CT machines commonly have the function to visualize 3D volumetric data, just like Fig. 18(B). However, it doesn't support data statistics or force analysis. Specially, the quantity of roadbed data is huge to deal with, intuitive interactivity and efficient computation are expected. According to the real demands of road CT dataset analysis, we roughly remove the pitch region by tuning the transfer function in volume rendering to obtain the multi-target segmentation masks, instead of using TurtleSeg. And the rest gravel shape modeling steps keep the same with this paper's steps. Thus, based on the 3D model produced by our method, we can easily accomplish such tasks.

7. Conclusions

In this paper, we have presented an interactive framework for the detail-preserving 3D shape implicit reconstruction from raw volumetric dataset. The newly-introduced Hessian constraints generalize the least squares RBF implicits, which can guarantee the reconstructed object to well respect high-order requirements. Meanwhile, many of the involved technical elements, including data specific importance sampling for adaptive spherical cover generation, determination of local samples for close sheets distinguishing, and the parallel solvers of local least squares systems, also contribute to many shape modeling related applications. Moreover, different types of carefully-designed experiments and practical applications have demonstrated our method's apparent advantages in terms of accuracy, efficiency, flexibility, and versatility.

However, at present our method still has some limitations. First, it is still a little complex for novice users to quickly place well-distributed cross-sectional constraints, because poorly-distributed constraints tend to cause unnecessary manipulation. Yet, the user could do it well after a brief training. In the future, we will further improve the convenience of our method by automatically providing some cues during user interaction. Besides, although solving local least squares systems in parallel could reduce temporal expenses to certain extent, to further enhance our method's time performance, we should design a CUDA-based parallel algorithm to simultaneously solve all the local least squares systems on graphics hardware. What's more, we will also verify our method over more field-specific volumetric datasets to further extend our method's application scope.

Acknowledgment

This research is supported in part by National Natural Science Foundation of China (No. 61190120, 61190121, 61190125, 61300067, 61672077, 61672149, and 61532002) and the National Key Research and Development Program of China (No. 2016YFC1000307).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.cag.2017.01.001.

References

[1] Owada S, Nielsen F, Igarashi T. Volume catcher. In: Proceedings of the 2005 symposium on Interactive 3D graphics and games. ACM; 2005. p. 111–16.
[2] Alexa M, Behr J, Cohen-Or D, Fleishman S, Levin D, Silva CT. Computing and rendering point set surfaces. IEEE Trans Vis Comput Graphics 2003;9(1):3–15.
[3] Ohtake Y, Belyaev A, Alexa M, Turk G, Seidel H-P. Multi-level partition of unity implicits. ACM Trans Graph 2003;22(3):463–70.
[4] Carr JC, Beatson RK, Cherrie JB, Mitchell TJ, Fright WR, McCallum BC, et al. Reconstruction and representation of 3D objects with radial basis functions. In: Proceedings of the 28th annual conference on computer graphics and interactive techniques. ACM; 2001. p. 67–76.
[5] Pan R, Meng X, Whangbo T. Hermite variational implicit surface reconstruction. Sci China Ser F: Inform Sci 2009;52(2):308–15.
[6] Li Y, Sun J, Tang C-K, Shum H-Y. Lazy snapping. ACM Trans. Gr. 2004;23(3):303–8.
[7] Macedo I, Gois JP, Velho L. Hermite radial basis functions implicits. Comput. Gr. Forum 2011;30(1):27–42.
[8] Li S, Yan D, Li X, Hao A, Qin H. Detail-preserving 3d shape modeling from raw volumetric dataset via hessian-constrained local implicit surfaces optimization. In: Proceedings of 2016 International Conference on Cyberworlds. IEEE; 2016. p. 25–32.
[9] Korfiatis P, Skiadopoulos S, Sakellaropoulos P, Kalogeropoulou C, Costaridou L. Combining 2D wavelet edge highlighting and 3D thresholding for lung segmentation in thin-slice CT, vol. 80. British Institute of Radiology; 2014. p. 996–1005.
[10] Celebi ME, Kingravi HA, Vela PA. A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert Syst. Appl. 2013;40(1):200–10.
[11] Anand S, Mittal S, Tuzel O, Meer P. Semi-supervised kernel mean shift clustering. IEEE Trans. Pattern Anal. Mach. Intell. 2014;36(6):1201–15.
[12] De Bruin PW, Dercksen VJ, Post FH, Vossepoel AM, Streekstra GJ, Vos FM. Interactive 3D segmentation using connected orthogonal contours. Comput. Biol. Med. 2005;35(4):329–46.
[13] Aliroteh M, McInerney T. Sketchsurfaces: Sketch-line initialized deformable surfaces for efficient and controllable interactive 3D medical image segmentation. In: Advances in Visual Computing. Springer; 2007. p. 542–53.
[14] Liu L, Bajaj C, Deasy J, Low DA, Ju T. Surface reconstruction from non-parallel curve networks. Comput. Graph. Forum 2008;27(2):155–63.
[15] Kass M, Witkin A, Terzopoulos D. Snakes: Active contour models. Int. J. Comput. Vis. 1988;1(4):321–31.
[16] Nguyen TNA, Cai J, Zhang J, Zheng J. Robust interactive image segmentation using convex active contours. IEEE Trans. Image Process. 2012;21(8):3734–43.
[17] Top A, Hamarneh G, Abugharbieh R. Active learning for interactive 3D image segmentation. In: Medical Image Computing and Computer-Assisted Intervention. Springer; 2011a. p. 603–10.
[18] Heckel F, Konrad O, Hahn HK, Peitgen H-O. Interactive 3D medical image segmentation with energy-minimizing implicit functions. Comput. Graphics Forum 2011;35(2):275–87.
[19] Wu Z. Compactly supported positive definite radial functions. Adv. Comput. Math. 1995;4(1):283–92.
[20] Wendland H. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. Adv. Comput. Math. 1995;4(1):389–96.
[21] Turk G, O'Brien JF. Shape transformation using variational implicit functions. In: Proceedings of the 26th annual conference on Computer graphics and interactive techniques, 20(3); 2005. p. 335–42.
[22] Ijiri T, Yokota H. Contour-based interface for refining volume segmentation. Comput. Gr. Forum 2010;29(7):2153–60.
[23] Ijiri T, Yoshizawa S, Sato Y, Ito M, Yokota H. Bilateral hermite radial basis functions for contour-based volume segmentation. Comput. Gr. Forum 2013;32(2):123–32.
[24] Top A, Hamarneh G, Abugharbieh R. TurtleSeg. http://www.turtleseg.org/.
[25] Huang H, Li D, Zhang H, Ascher UM, Cohen-Or D. Consolidation of unorganized point clouds for surface reconstruction. ACM Trans. Gr. 2009;28(5):89–97.
[26] Liu J, Cao J, Liu X, Wang J, Wang X, Shi X. Mendable consistent orientation of point clouds. Comput.-Aided Des. 2014;55:26–36.
[27] Ohtake Y, Belyaev A, Seidel H-P. An integrating approach to meshing scattered point data. In: Proceedings of the symposium on Solid and physical modeling. ACM; 2005. p. 61–9.
[28] Guennebaud G, Jacob B, Avery P, Bachrach A, Barthelemy S, et al.. Eigen v3; 2010. http://eigen.tuxfamily.org.
[29] Ho C-C, Wu F-C, Chen B-Y, Chuang Y-Y, Ouhyoung M. Cubical Marching Squares: Adaptive feature preserving surface extraction from volume data. Comput. Gr. Forum 2005;24(3):537–45.