Detail-preserving 3D Shape Modeling from Raw Volumetric Dataset via Hessian-constrained Local Implicit Surfaces Optimization

Shuai Li
State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, China

Dehui Yan
Beihang University, China

Xiangyang Li
Beihang University, China

Aimin Hao*
State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, China, Email: ham@buaa.edu.cn

Hong Qin
Department of Computer Science, Stony Brook University, USA, Email: qin@cs.stonybrook.edu

***Abstract*—Massive routinely-acquired raw volumetric datasets are hard to be deeply exploited by cyber worlds related 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. 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. All the results demonstrate our method's advantages in the accuracy, detail-preserving, efficiency, and versatility of shape modeling.**

***Keywords*-3D shape modeling; raw volumetric dataset; implicit surfaces; Hessian constraints**

I. Introduction and Motivation

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.

The volume segmentation approaches can be classified into clustering based methods and interaction based methods. Although clustering based methods have advantages in efficiency and 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.

Benefitting from user-specified contours, the interactive shape modeling methods usually involve a surface 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 have well-known advantages in high accurate representation and low computation cost. However, they tend to produce deformed messy shapes when handling low-quality data with noise, outliers, and uneven sampling density. Global implicit methods, including RBF based approaches [2], variational approach [3], graph-cut approach [4], and Hermite RBF Implicits (HRBF) [5], 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 least squares implicits based on Hessian constraints and incorporating them into iterative optimization of local implicit surfaces. 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 RBF implicit by incorporating the second-derivative Hessian constraints, which facilitates detail-preserving implicit surface reconstruction over the jagged boundary voxels of the segmented objects.
- We design an adaptive spherical cover generation scheme to adaptively determine the supporting do-

978-1-5090-2303-5/16 $31.00 © 2016 IEEE
DOI 10.1109/CW.2016.12

main 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.

II. Related Work

Interactive Contour Assisted Segmentation. Considering that fully-manual segmentation process is laborious and time-consuming, while fully-automatic algorithms 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. [6] proposed a method to generate boundary surfaces by directly connecting well-organized contour vertices. Aliroteh et al. [7] 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. [8] initialized boundary surfaces using a Voronoi diagram based algorithm and then smoothed them. Motivated by physics-based deformable models, Kass et al. [9] proposed the active contour method, and its many extensions have also been developed [10]. 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 [11] 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 [12], and then extract the zero iso-surface. Due to its potential to satisfactorily handle sparse sample points, great attention has been paid to RBF [2]. 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 normal into the problem formulation, Pan et al.[3] proposed Hermite variational implicit surface, and Macedo et al.[5] proposed Hermite Radial Basis Function (HRBF), which is a particular case of Hermite-Birkhoff interpolation with RBF. Wu [13] and Wendland [14] 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 [2], [15] are less sensitive to the quality of input data, they require the solving of large and dense matrix system. Besides, Ijiri et al. [16] proposed a system to refine volumetric shape surface by piecewise fine-tuning curved contours, and they also introduced a new implicit method [17] 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.

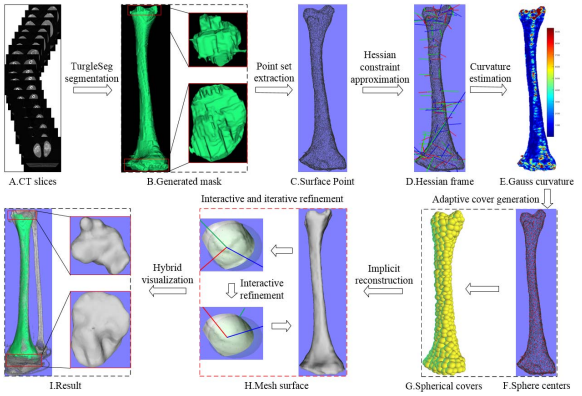

Figure 1. Flow chart of our method. (A) Original CT slices; (B) The segmentation mask generated by TurtleSeg; (C) The extracted surface points; (D) The Hessian frames; (E) Gaussian curvature; (F,G) Spherical cover generation for local least-squares implicit surface reconstruction; (H) The iterative refinement based on user-controllable Hessian constraints; (I) The finally-reconstructed shape.

III. Method Overview

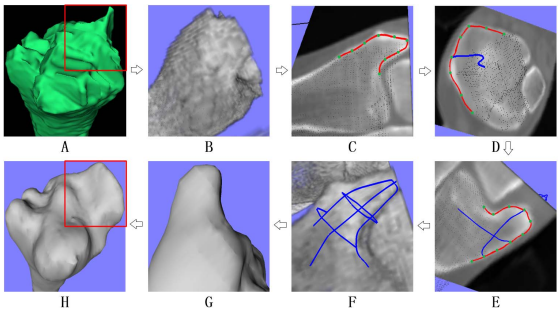

Figure 2. Illustration of the interactive constraint manipulation. (A) The rough segmentation mask with obvious errors; (B) The potential surface points exacted from the segmentation mask; (C,D,E, and F) Interactively placing contours on cross sections; (G,H) The final shape modeling results.

Our 3D shape 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 [18] 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 (Fig. 2(C) $\sim$ Fig. 2(F)). As shown in Fig. 2, this process is simple and does not require much expert knowledge.

Fig. 1(D) demonstrates the frames of Hessian matrix on some points, and we conduct local quadric least squares fitting to approximate the Hessian matrix (see Section IV-B 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)-(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. The red box in Fig. 1 illustrates our interactive and iterative manipulation procedures, and we can locally edit the Hessian constraints and update the scalar field until we are satisfied with the results (Fig. 1(I)).

IV. 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.

A. 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 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 Possion disk sampling to perform down-sampling. Considering the boundaries may be jagged, we further adopt WLOP to reduce noise and outliers of the potential surface point set. At last, we employ the method proposed by Lie et al. [19] 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.

B. Initialization of Hessian Constraints

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 $\mathbf{x}_i$.

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

$$\arg\min_{\mathbf{q}} \sum_{\mathbf{x}_k \in nei(\mathbf{x}_i)} (G_i(\mathbf{x}_k)^2 + \parallel \nabla G_i(\mathbf{x}_k) - \mathbf{n}_k \parallel^2), \quad (2)$$

where $\mathbf{q} = (\gamma_1, ..., \gamma_{10})$ 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:

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

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

C. 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 IV-D for details), the parameters and weight are determined by Hessian-constrained local least squares RBF implicits (see Section V-A 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.

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

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 $\parallel \mathbf{p}_j - \mathbf{c} \parallel \leq r$, $w(x)$ 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 $\parallel \mathbf{x} - \mathbf{c} \parallel \leq r$ and L is the main diagonal length of the bounding box of point set $\mathcal{P}$, and please refer to [20] for more details. 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

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

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.

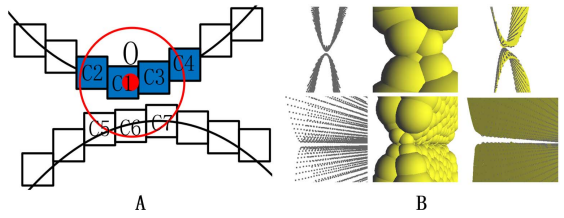

Figure 3. Illustration of separable close sheets in a toy example.

D. 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. 3(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 convert these cells into an adjacency graph. Here the adjacency is defined as the one-ring neighbor with similar normal orientation. Taking Fig. 3(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, which is covered by s_i and has the similar orientation with the point in $\mathcal{S}_i$, within the cell $C1$ into $\mathcal{S}_i$. 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. 3(B) illustrates this process in a toy example.

V. Interactive and Iterative Implicit Reconstruction

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 by editing the Hessian constraints interactively and updating the scalar field iteratively.

A. 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 multi-quadric $\phi(r) = \sqrt{r^2 + c^2}$ (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, our reconstruction problem within each sphere s_l, is converted into an optimization problem as follows,

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

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

where $p_j(\mathbf{x})$ is the linear polynomials, $\mathbf{q} = (\alpha_1, ..., \alpha_m, \beta_1, ..., \beta_4)$ 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 represents 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 $\lambda_1, \lambda_2, \lambda_3$ (satisfying $\lambda_1 + \lambda_2 + \lambda_3 = 1$) control the importance of different items. The system can be solved by a least squares solver.

B. Global Blending of Local Implicits

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. 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 $\| \mathbf{x} - \mathbf{c}_i \| \leq r_i$.

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.4 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.4(C) and Fig.4(D) proves the effectiveness of our global blending procedure. Fig.5 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.

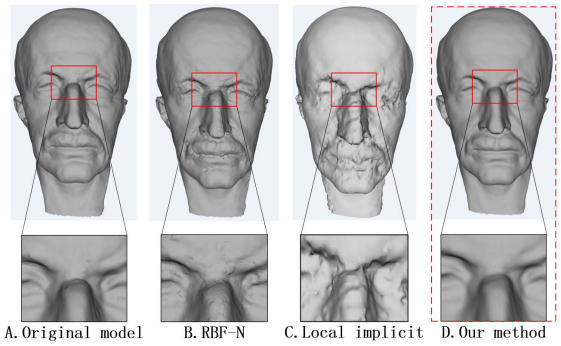

Figure 4. The comparison of implicit shape modeling results between our method and RBF-N method over Max-Planck model. (A) shows the original model, (B) shows the RBF-N result, (C) shows the intermediate result of our method before global blending, and (D) shows our final result.

C. 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 $\mathbf{H}_i = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{-1}$, where $\mathbf{P} = (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$, $\mathbf{\Lambda} = diag(v_1, v_2, v_3)$ 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

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

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.

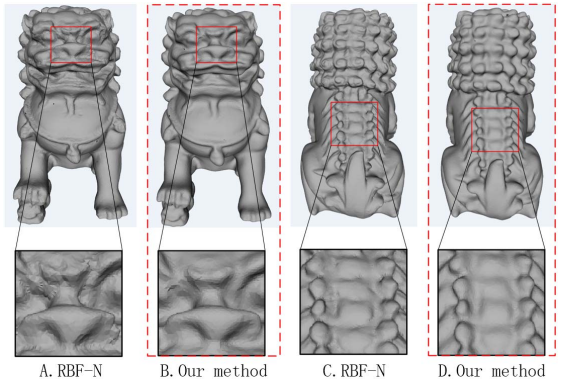

Figure 5. The comparison of implicit shape modeling results between our method and RBF-N method over a Lion model. The first, third columns show the results of RBF-N method, and the second, fourth columns show our results.

We can iteratively repeat the refinement process until we are satisfying with the results. Fig. 6 shows the reconstruction results with different Hessian-constraints via interactive manipulation.

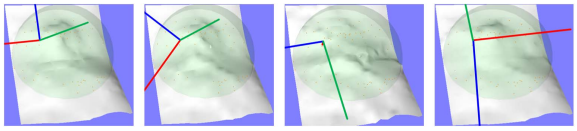

Figure 6. The shape modeling results under different Hessian constraints and the light green color indicates the controlling sphere.

D. Numerical Implementation

To handle the time-consuming computation of Hessian constraints involved local least squares RBF implicits, we implement our method in a parallel way by assigning a well-designed OpenMP thread to each spherical cover. To achieve this goal, we carefully design the structure of the spherical cover by storing the samples of each sphere individually. According to Eq. 7 and Eq. 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

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

Algorithm 1: 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 $\mathbf{A}$ and vector $\mathbf{y}$ according to Eq. 13,
3. Use the Eigen library [21] to solve the least squares system $\mathbf{A}\mathbf{x} = \mathbf{y}$.
}

Following the pipeline in Fig. 1, we detail the overall numerical implementation of our prototype system in Algorithm 2. 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 entire mesh of an object, 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 partial mesh, the region is set to be the bounding box

Algorithm 2: Numerical Implementation of Interactive 3D Shape Modeling.

input : The segmentation mask volume.
output: The 3D shape modeling result.
1. Point set preprocessing
Including potential surface point set extraction, Possion disk
sampling, WLOP, normal estimation and Hessian constraints
approximation;
2. Adaptive spherical cover generation
3. Solve local least squares systems parallelly
Determine local samples,
Invoke Algorithm 1 for all spherical covers;
4. Interactive and iterative surface refinement
Extract global iso-surface,
while(not satisfying 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
invoking Algorithm 1,
(5) Update global iso-surface.
}

of the controlling sphere, and the resolution is set to be $16 * 16 * 16$.

VI. 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 I shows the experimental data statistics in details. The resolution of volume 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 of 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 system.

Fig. 7 and Fig. 9 compare the shape modeling results among RBF-N based method, HRBF based method, and our method. In nature, the HRBF based method is an interpolation method, which is sensitive to noise when handling samples with outliers (Fig. 8). From (Fig. 7(B)

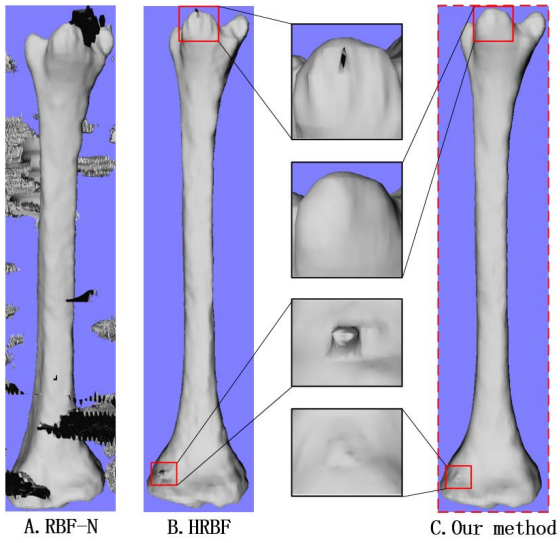

Figure 7. The segmented surface comparison among RBF-N based method, HRBF based method, and our method over the tibia volume.

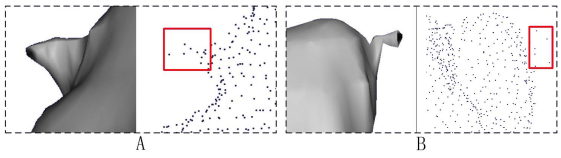

Figure 8. The artifacts of Fig. 7(B) are shown from another view point, and the point set visualization of the corresponding regions.

and Fig. 9(B), we can see the segmented surfaces by the HRBF based method exhibit many obvious artifacts. Meanwhile, Fig. 7(A) and Fig. 9(A) show that the segmented surfaces by the RBF-N based method also contain some artifacts caused by the inaccurate scalar field. In sharp contrast, our method can produce a smooth shape surface. Meanwhile, Table II documents the statistics of error approximation on different models. Those from the RBF-N method are produced with $\lambda_1 = 0.9, \lambda_2 = 0.1$. 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.

Fig. 10 and Fig. 11 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 to place 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 the quality of 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. From the

Table I
THE EXPERIMENTAL PERFORMANCE STATISTICS (IN SECONDS).

Dataset	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

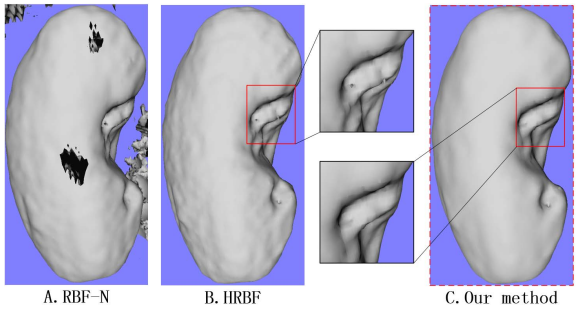

Figure 9. The reconstructed surface comparison among RBF-N based method, HRBF based method, and our method in the kidney example.

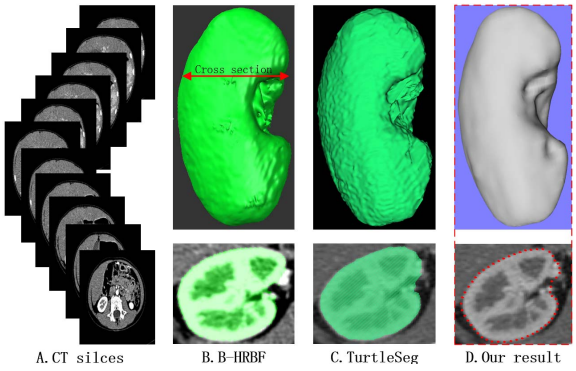

Figure 10. Comparison with the B-HRBF method, TurtleSeg method, and ours in the kidney example. (A) shows the input CT slice data; (B) shows the result generated by the B-HRBF method; (C) shows the result of TurtleSeg; (D) shows our shape modeling result. The second row shows the details on the cross-section slices.

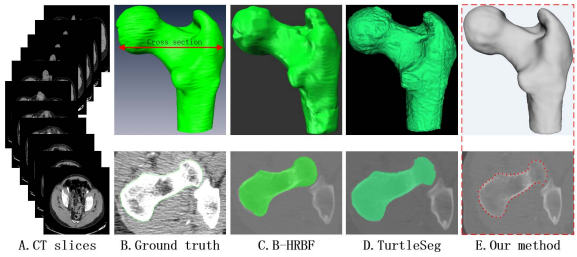

Figure 11. Comparison with the ground truth, the B-HRBF method, TurtleSeg method and ours in the femur example. (A) shows the input CT slice data; (B) shows the ground truth manually generated by an expert; (C) shows the result generated by the B-HRBF method; (D) shows the result of TurtleSeg; (E) shows our shape modeling result. The second row shows the details on the cross-section slices.

results shown in the cross-section slices, although such three methods can produce similar results, benefitting from the full use of the second-order information, our method achieves more accurate shape modeling results with more high-fidelity details. Therefore, all the aforementioned experiments have demonstrated our method's advantages in accuracy, detail preservation, and high-order smoothness.

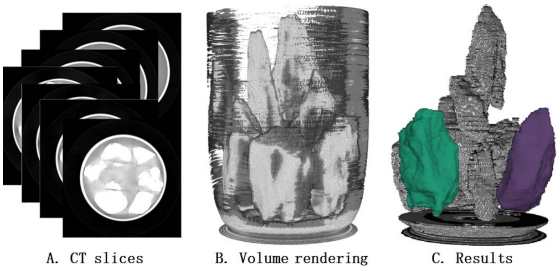

Figure 12. The multi-target shape modeling results of our method in a roadbed example. (A) Original CT slices; (B) Direct volume rendering; (C) The multi-target shape modeling result.

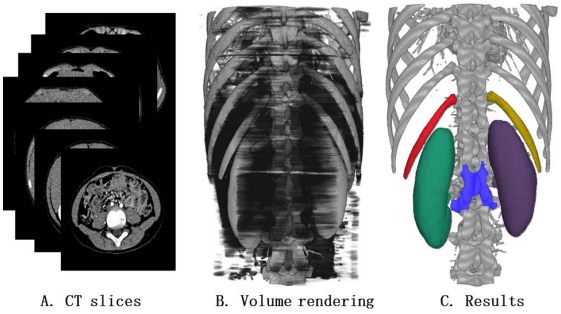

Figure 13. The multi-target shape modeling results of our method in a digital human example. (A) Original CT slices; (B) Direct volume rendering; (C) The multi-target shape modeling results.

Table II
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**

Besides, Fig. 12 and Fig. 13 respectively show the multi-target shape modeling results of our method in a roadbed example and a digital human example, wherein

the colored objects in the sub-figures (C) represent the surface meshes of the segmented targets while other objects are visualized directly using the ray-casting based volume rendering method. It clearly shows our method's versatility in practical applications.

VII. Conclusions

In this paper we have presented an interactive framework for the detail-preserving 3D shape modeling from raw volumetric dataset. The newly-introduced Hessian constraints generalize the least squares RBF implicits, which can guarantee the segmented object to 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 in order to separate close sheets, and parallel solvers of local least squares systems, also contribute to many cyber world related applications. Moreover, different types of carefully-designed experiments have demonstrated our method's apparent advantages in terms of accuracy, efficiency, flexibility, and versatility.

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 drastically enhance our method's time performance, we should design a GPU-based parallel algorithm to simultaneously solve all the local least squares systems in graphics hardware.

Acknowledgment

This research is supported in part by National Natural Science Foundation of China (No. 61190120, 61190121, 61190125, 61300067, and 61532002).

References

[1] S. Owada, F. Nielsen, and T. Igarashi, "Volume catcher," in *Proceedings of the 2005 symposium on Interactive 3D graphics and games*. ACM, 2005, pp. 111–116.

[2] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans, "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, pp. 67–76.

[3] R. Pan, X. Meng, and T. Whangbo, "Hermite variational implicit surface reconstruction," *Science in China Series F: Information Sciences*, vol. 52, no. 2, pp. 308–315, 2009.

[4] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum, "Lazy snapping," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 303–308, 2004.

[5] I. Macedo, J. P. Gois, and L. Velho, "Hermite radial basis functions implicits," *Computer Graphics Forum*, vol. 30, no. 1, pp. 27–42, 2011.

[6] P. W. De Bruin, V. J. Dercksen, F. H. Post, A. M. Vossepoel, G. J. Streekstra, and F. M. Vos, "Interactive 3D segmentation using connected orthogonal contours," *Computers in Biology and Medicine*, vol. 35, no. 4, pp. 329–346, 2005.

[7] M. Aliroteh and T. McInerney, "Sketchsurfaces: Sketch-line initialized deformable surfaces for efficient and controllable interactive 3D medical image segmentation," in *Advances in Visual Computing*. Springer, 2007, pp. 542–553.

[8] L. Liu, C. Bajaj, J. Deasy, D. A. Low, and T. Ju, "Surface reconstruction from non-parallel curve networks," *Computer Graphics Forum*, vol. 27, no. 2, pp. 155–163, 2008.

[9] M. Kass, A. P. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International journal of computer vision*, vol. 1, no. 4, pp. 321–331, 1988.

[10] N. T. N. Anh, J. Cai, J. Zhang, and J. Zheng, "Robust interactive image segmentation using convex active contours," *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3734–3743, 2012.

[11] A. Top, G. Hamarneh, and R. Abugharbieh, "Active learning for interactive 3D image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*. Springer, 2011, pp. 603–610.

[12] F. Heckel, O. Konrad, H. K. Hahn, and H.-O. Peitgen, "Interactive 3D medical image segmentation with energy-minimizing implicit functions," *Computer Graphics Forum*, vol. 35, no. 2, pp. 275–287, 2011.

[13] Z. Wu, "Compactly supported positive definite radial functions," *Advances in Computational Mathematics*, vol. 4, no. 1, pp. 283–292, 1995.

[14] H. Wendland, "Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree," *Advances in computational Mathematics*, vol. 4, no. 1, pp. 389–396, 1995.

[15] G. Turk and J. F. O'Brien, "Shape transformation using variational implicit functions," *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, vol. 20, no. 3, pp. 335–342, 2005.

[16] T. Ijiri and H. Yokota, "Contour-based interface for refining volume segmentation," *Computer Graphics Forum*, vol. 29, no. 7, pp. 2153–2160, 2010.

[17] T. Ijiri, S. Yoshizawa, Y. Sato, M. Ito, and H. Yokota, "Bilateral hermite radial basis functions for contour-based volume segmentation," *Computer Graphics Forum*, vol. 32, no. 2, pp. 123–132, 2013.

[18] A. Top, G. Hamarneh, and R. Abugharbieh, "TurtleSeg," http://www.turtleseg.org/, 2011.

[19] J. Liu, J. Cao, X. Liu, J. Wang, X. Wang, and X. Shi, "Mendable consistent orientation of point clouds," *Computer-Aided Design*, vol. 55, pp. 26–36, 2014.

[20] Y. Ohtake, A. G. Belyaev, and H.-P. Seidel, "An integrating approach to meshing scattered point data," in *symposium on Solid and physical modeling*. ACM, 2005, pp. 61–69.

[21] G. Gaël and J. Benoît, "Eigen v3," http://eigen.tuxfamily.org, 2010.