IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 6, 2019, accepted October 17, 2019, date of publication October 21, 2019, date of current version November 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2948621

A Fast 3D Brain Extraction and Visualization Framework Using Active Contour and Modern OpenGL Pipelines

JUNCHEN WANG[1,2], ZHEN SUN[1], HONGLEI JI[1], XIAOHUI ZHANG[1], TIANMIAO WANG[1,2], AND YU SHEN[1]

[1]School of Mechanical Engineering and Automation, Beihang University, Beijing 100191, China
[2]Beijing Advanced Innovation Center for Biomedical Engineering, Beihang University, Beijing 100083, China

Corresponding author: Yu Shen (yu_shen@buaa.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61701014 and Grant 61911540075, and in part by the National Key Research and Development Program of China under Grant 2017YFB1303004.

ABSTRACT Brain extraction is a process of removing non-brain tissue in the brain magnetic resonance (MR) images and serves as a first step towards more delicate brain segmentation. Although many brain extraction methods have been proposed in the literature, most of them are either laborious or time consuming, and lack of instant visualization. This leads to a time lag between image acquisition and comprehensive visualization. Especially for intraoperative image based neurosurgery navigation, the time lag from image acquisition to brain visualization should be reduced as much as possible. In this paper, we propose an end-to-end fast brain extraction and visualization framework. The input is a T1-weighted MR volume and the output is comprehensive brain visualization. An improved brain extraction tool (BET) algorithm is proposed to evolve a 3D active mesh model to fit the brain surface in the 3D image. Then the brain mask is generated per slice using a polygon fill algorithm. At last, a ray-casting volume rendering algorithm is used to visualize the brain surface with the help of the generated mask. All the operations are performed using the modern OpenGL pipelines running on a graphics processing unit (GPU). Experiments were performed on two publicly available datasets and one clinical dataset to compare our method with five state-of-the-art methods including the original BET in terms of segmentation accuracy and time cost. Our method achieved mean Dice coefficients of 96.8%, 97.1%, 98.5% and mean time cost of 361 ms, 341 ms, 502 ms on the three datasets, outperforming all the other methods.

INDEX TERMS Brain extraction, image segmentation, active contour, visualization, neurosurgery.

I. INTRODUCTION

Brain extraction (also known as skull stripping) is a first step towards brain image analysis and is widely performed prior to more delicate brain segmentation [1]. It refers to the process of removing non-brain tissue such as skull, dura, and scalp in the brain magnetic resonance (MR) images. It is also regarded as the brain/non-brain segmentation in which the brain mainly consists of white matter (WM), gray matter (GM), cerebrospinal fluid (CSF), and cerebellum. After brain extraction, brain model generation and/or fine brain tissue classification (e.g., WM, GM, CSF, cerebellum) can be performed to facilitate the following brain image analysis which is eventually employed for brain disease diagnosis, therapy, and prediction [2]–[5].

In neurosurgery navigation [6]–[8], the brain structure from MR images is visualized as the atlas to guide the interventional procedure which requires brain extraction to be performed in advance. In addition, the details of the brain structure such as cerebral cortex, vessels, sulci and gyri are supposed to be vividly visualized so that the surgeon could make appropriate interventional path. In general, brain segmentation is a challenging problem due to the nonhomogeneous intensity distribution of brain structures and the intensity overlap with surrounding non-brain structures, calling for sophisticated algorithms and significant human input for parameter tunning. High inter-subject variability across individual brains and presence of pathological regions further

The associate editor coordinating the review of this manuscript and approving it for publication was Gustavo Olague.

reduce the generalization of the algorithms. Although many brain/non-brain segmentation methods exist, most of them are labour-intensive and time consuming and lack of instant brain visualization. The output of existing brain segmentation methods needs further to be processed (e.g., 3D mesh generation, reduction, smoothing) for the final visualization. The visualization cannot reflect the change of segmentation results in real time. For intraoperative MR image based neurosurgical navigation [9]–[11], it is preferred that the time lag from image acquisition to fine visualization should be reduced as much as possible.

In this paper, we are not addressing the brain segmentation problem at the finest level, instead we focus on the integrated framework of fast whole brain extraction and accurate vivid visualization using modern OpenGL pipelines running on a graphics processing unit (GPU).

II. RELATED WORK

Existing brain extraction methods can be divided into the following four categories: low-level image processing with histogram analysis and connected component filtering [12]–[19], atlas matching [20]–[24], active contour models [25]–[30], and learning-based approaches [31]–[35]. Each of the existing methods has strengths and weaknesses. Most of the methods work in a 2D manner (i.e., segmentation is performed slice by slice). In addition, some hybrid methods that combine at least two of aforementioned methods have also been proposed [36]. For example, atlas model and active contour model are often combined to make use of both local image features and prior global shape information to achieve better segmentation results [37], [38].

Low-level image processing methods isolate the brain region in the axial slice by carrying out thresholding or edge detection followed by a series of morphology operations to break the thin bridging area between brain and non brain tissues. Then a connected component filter is applied to extract the isolated brain region. Preprocessing such as anisotropic diffusion filtering may help in improving the brain isolation. Instead of applying morphology operations, Sadananthan *et al.* propose to use graph cuts to isolate the brain region [39]. The disadvantages of these low-level processing based methods include high sensitivity to signal noise ratio (SNR) and edge discontinuity, difficult parameter tuning, and low robustness against different imaging protocols.

Atlas matching methods first construct manually labeled brain atlases and then match the atlases against the target data volume using a deformable image registration algorithm. The labels are therefore mapped to the target volume space either simply using the resulting transformation, or through a more complicated label fusion procedure [40]. Apart from high computational cost, atlas based segmentation methods suffer from registration failure and misalignment if an inappropriate registration model is chosen or the target volume lies far from the atlas population.

Active contour models are frequently used in image segmentation tasks to find the boundary of region of interest (ROI). The core idea is to minimize a functional defined by the contour representation and image features. The minimization is eventually converted to solving partial differential equations using Euler equations. The resulting behavior is that the contour is being attracted onto the image's salient edges. According to the representation of the contour, the active contour model has two main forms: snakes [41] and geodesic active contours [42]. Compared with snakes, geodesic active contours use the level set of a higher dimensional function to represent the contour which can naturally handle curve splitting and merging. Active contour models are driven by local image features and may penetrate into the boundary if the local edge feature is not salient. Besides, an initial contour should be placed nearby the target boundary as the start point of contour evolution. To overcome this limitation, Leventon *et al.* propose to incorporate prior shapes into the framework of geodesic active contours [26]. However, the prior shapes suffer from inter-subject variation across different individual brains. The trade-off between the global shape guidance and the local feature fidelity is difficult to balance in a real application.

Learning-based approaches include parameter learning [32], hidden Markov model [31], generative and discriminative model [43], and convolutional networks [33]–[35]. Recently, deep convolutional networks have gained great success and outperform traditional methods in various image segmentation tasks. In the brain segmentation, deep convolutional networks are mainly proposed to perform fine tissue classification after brain extraction [34]. Hwang *et al.* employ 3D U-Net [44] for skull stripping, however, no quantitative evaluations were presented. The performance of learning-based methods heavily rely on the training data and they perform well mostly on the specific dataset on which the learner is trained or tuned. It is difficult to obtain adequate well labeled datasets which can reflect the variability across large-scale individual brains.

It has been realized that every method has both strengths and weaknesses, and no single method generalizes well for a large scale of datasets. Local feature based methods have difficulty in parameter tuning and suffer from imaging noise. Statistical model/learning based methods struggle with the model generalization and local fidelity. Our method is based on the BET algorithm [28] which is a local feature based method with the advantages of being computationally efficient and less sensitive to parameter choice. Our contribution has two folds: 1. as the methodology innovation we have modified and improved the original BET algorithm to increase the segmentation accuracy of cerebral cortex with deep sulci and gyri; 2. as the practical implementation we have designed and implemented a very fast GPU-based parallel processing pipeline for the improved BET and instant comprehensive brain visualization. Instant visualization is very important for quickly and intuitively tuning parameters of segmentation algorithms [45].

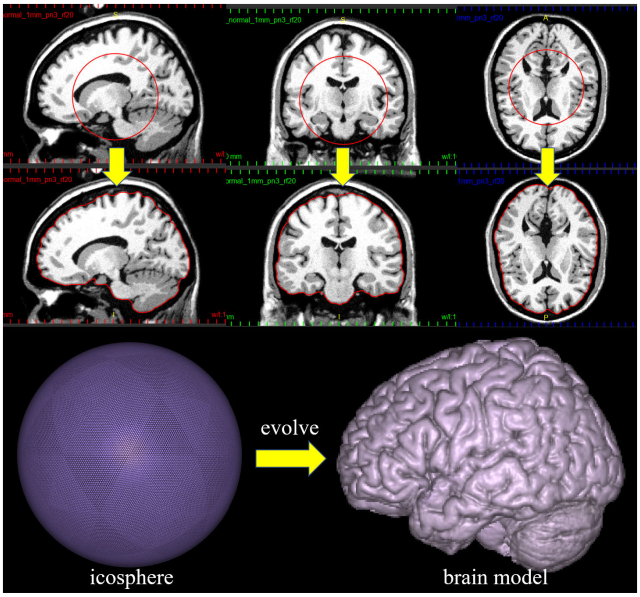


FIGURE 1. Brain extraction using BET algorithm.

III. METHOD

A. IMPROVED BET

For the details of the BET algorithm we refer readers to the original paper [28]. Here we only give necessary description of BET and conclusive formula for the understanding of our approach. The BET algorithm evolves an icosphere to fit the brain surface in the MR volume as shown in Fig. 1. The icosphere is generated by tessellating an icosahedron and is coarsely placed in the center of the brain. At each iteration of the evolution, the displacement vector $\boldsymbol{u}$ of each vertex of the icosphere is composed of three terms:

$$\boldsymbol{u} = 0.5\boldsymbol{s}_t + f_2\boldsymbol{s}_n + 0.05lf_3\boldsymbol{n} \tag{1}$$

which control the within-surface vertex spacing, surface smoothness and brain surface attraction, respectively. $\boldsymbol{n}$ is the unit normal vector of the local surface at the vertex in question (current vertex); l is the mean distance from the current vertex to its adjacent vertices; and $\boldsymbol{s}_n = (\boldsymbol{n} \cdot \boldsymbol{s})\boldsymbol{n}$, $\boldsymbol{s}_t = \boldsymbol{s} - \boldsymbol{s}_n$ where $\boldsymbol{s}$ is the vector pointing from the current vertex to the mean position of its adjacent vertices. Figure 2 illustrates the relationships between $\boldsymbol{s}_n$, $\boldsymbol{s}_t$, $\boldsymbol{s}$ and $\boldsymbol{n}$.

Different with the original algorithm, we propose to calculate f_2 using a sigmoid function:

$$f_2 = \frac{1}{1 + e^{\frac{r-\beta}{\alpha}}} \tag{2}$$

$$r = \frac{l^2}{2|\boldsymbol{s}_n|} \tag{3}$$

where r approximates the local radius of curvature; α and β are smoothness parameters empirically taking 1 mm and 8 mm respectively. Only β needs to be adjusted adaptively in rare cases where self-intersection happens during the surface evolution. Increasing β will prevent self-intersection. f_2 can be seen as an internal force that make the evolving surface smooth.

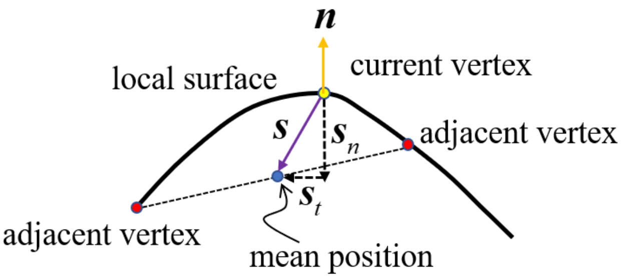


FIGURE 2. Relationships between s_n, s_t, s and n.

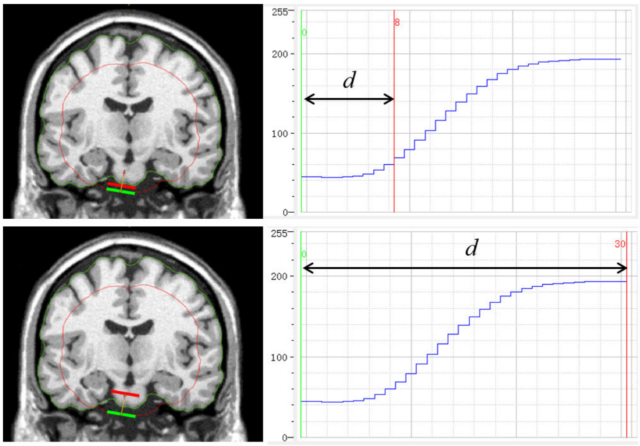


FIGURE 3. Choice of d affects the direction of the image force.

f_3 is given by:

$$f_3 = \frac{2(I_{\min} - t_1)}{I_{\max} - t_2} \tag{4}$$

$$t_1 = (I_{\max} - t_2)b_t + t_2 \tag{5}$$

$$I_{\min} = \max(t_2, \min(t_m, I(0), I(1), \cdots, I(d))) \tag{6}$$

$$I_{\max} = \min(t_m, \max(t, I(0), I(1), \cdots, I(d))) \tag{7}$$

$$t = 0.9t_2 + 0.1t_{98} \tag{8}$$

where t_2 and t_{98} are the 2- and 98-percentile intensities of the histogram of the brain MR volume; t_m is the median intensity within the initial icosphere; b_t is an input parameter usually taking the value of 0.5; $I_{\min}$ and $I_{\max}$ are the robust estimates of minimum and maximum intensities of the line profile along the negative norm vector with the intensity function $I(i) = I(\boldsymbol{v} - i\boldsymbol{n})$, $i = 0, 1, \cdots, d$ where $\boldsymbol{v}$ represents the current vertex; d is the intensity searching depth which is recommended to take 20 mm in the original algorithm. Therefore the update position of the current vertex $\boldsymbol{v}^k$ at the step k is given by:

$$\boldsymbol{v}^{k+1} = \boldsymbol{v}^k + \boldsymbol{u}^k \tag{9}$$

f_3 can be seen as an image force that makes the evolving surface stick to the brain surface in the MR volume. It is determined by the line intensity profile starting at the current vertex and going along the negative normal vector into the

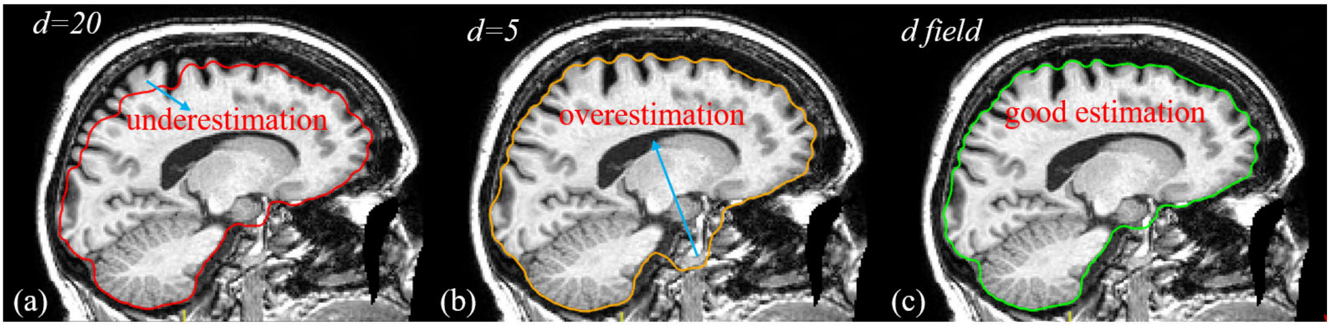


FIGURE 4. Constant d leads to problematic brain extraction. (a) and (b) overestimation and underestimation with constant d. (c) Improved extraction using the proposed d field.

depth of d. Let us assume $t_2 = 0$, $b_t = 0.5$ then we have $f_3 = (2I_{\min} - I_{\max})/I_{\max}$. If the line profile up to d is approximately a horizontal line ($I_{\min} \approx I_{\max}$), f_3 approaches one leading to a surface expansion. Otherwise if a step is present, f_3 approaches minus one leading to a surface contraction. With a more intuitive explanation, horizontal profile means tissue homogeneity indicating that the current vertex is still in the brain part while a step profile means CSF is encountered indicating the current vertex is running out of the brain part. The choice of d is important because the larger it is, the more likely a step is present. As shown in Fig 3, a smaller d makes the local surface move outwards while a larger d makes the local surface move inwards under the force of f_3. To conclude, a larger d tends to underestimate brain tissue (conservative) while a smaller d tends to overestimate brain tissue (aggressive).

In the original BET algorithm, d remains constant everywhere once specified. As illustrated in Fig. 4(a) and (b), we found that the constant d led to problematic brain extraction. Because in the region of cerebral cortex with deep sulci and gyri, a smaller d is desired for the evolving surface to penetrate CSF between GM. While in the region of pituitary and cerebellum, a larger d is desired for robust tissue discrimination. We therefore make a hypothesis that varying d according to different surface locations may overcome this problem. In this case, d actually becomes a scalar field defined on the evolving surface. As shown in Fig 5(a), note that in the region of cerebral cortex the brain's local shape has relatively high curvature (i.e., small radius of curvature). We propose to perform a sigmoid nonlinear mapping from the local radius of curvature r to the depth d.

$$d = (d_{\max} - d_{\min})\frac{1}{1 + e^{-\frac{r-b}{a}}} + d_{\min} \tag{10}$$

where $d_{\max}$ and $d_{\min}$ are the maximum and minimum d to be mapped to; a and b are parameters that clamp the effective mapping range of r to be $[b - 3a, b + 3a]$. We found that setting $a = 1$ mm and $b = 8$ mm works well for almost all cases. $d_{\min}$ is empirically chosen to be 1 mm and $d_{\max}$ is given according to the region division described as follows. As shown in Fig 5(b), assume that the brain volume has been oriented in the LPS (Left-Posterior-Superior) coordinate system whose origin is located at the center of gravity (COG) of the brain. $d_{\max}$ is set to be d_2 for those vertices of the evolving surface whose projection in the sagittal plane fall into the pole angle range of $[\theta, \pi]$, and to be d_1 for the rest vertices. θ is an input parameter that controls the region where a smaller d_2 is desired and is set to be 0 by default. d_1 and d_2 are by default set to be both 20 mm. For those brains which has deep sulci and gyri as shown in Fig. 4, d_2 is usually decreased to 5 mm. With the proposed d field, the problematic brain extraction in the original BET could be improved as shown in Fig. 4(c).

By testing on many datasets we have found some parameters can be fixed after optimizing, therefore in our approach we have only three parameters that are available for user tuning: d_2, b_t, and θ. Compared with the original BET that has only one input parameter b_t, the extra two input parameters

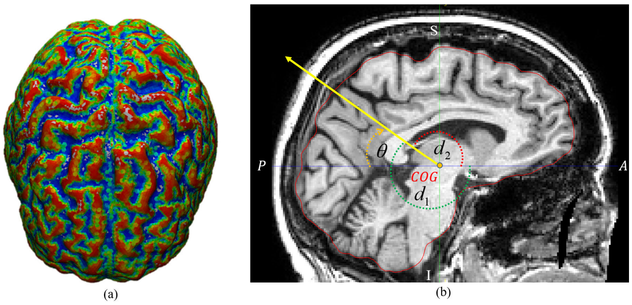


FIGURE 5. Nonlinear mapping from r to d. (a) Mean curvature distribution of the brain surface. (b) Region division of $d_{\max}$.

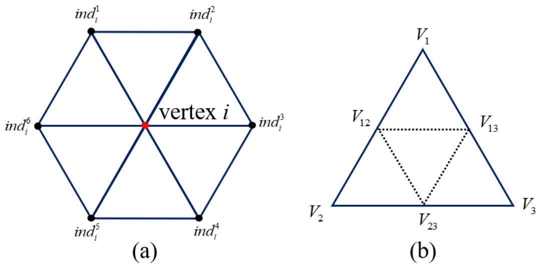
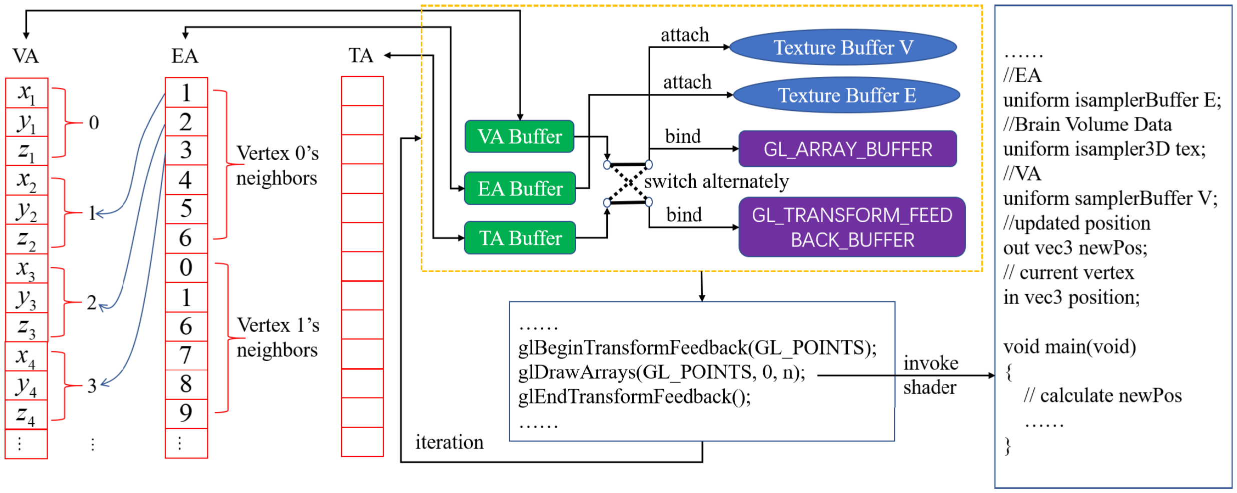


FIGURE 6. Topology of the surface. (a) Local connectivity. (b) Triangle tessellation. Note that the tessellated vertices are not in the same plane as V_1-V_2-V_3.



```
……
glBeginTransformFeedback(GL_POINTS);
glDrawArrays(GL_POINTS, 0, n);
glEndTransformFeedback();
……
```

```
……
//EA
uniform isamplerBuffer E;
//Brain Volume Data
uniform isampler3D tex;
//VA
uniform samplerBuffer V;
//updated position
out vec3 newPos;
// current vertex
in vec3 position;

void main(void)
{
  // calculate newPos
  ……
}
```

FIGURE 7. OpenGL pipeline for surface evolution.

d_2 and θ control the d-field for fine estimation in the region of cerebral cortex.

B. DATA STRUCTURE AND PARALLEL PROCESSING

Surface evolution can be inherently implemented in parallel by exploiting vertex shaders in modern OpenGL. The shader program is executed on all vertices of the surface. In each vertex shader instance, a vertex's position is updated and many shaders are invoked at the same time on a GPU so that the surface evolves in a parallel way. The updated surface vertices in the shaders are captured back using OpenGL's transform feedback mechanism and are used in the next iteration.

We represent the evolving surface using two arrays: vertex array VA and connectivity array EA. VA stores vertex coordinates in the order of $(x_i, y_i, z_i), i = 1, \cdots, n$ where n is the number of vertices and has the form of $n = 20 \cdot 4^m$ with m being the number of tessellation times. Each vertex can be referenced by its index in VA. EA stores the ordered indices of each vertex's neighboring vertices and has the form of $(ind_i^1, ind_i^2, ind_i^3, ind_i^4, ind_i^5, ind_i^6), i = 1, \cdots, n$ where ind_i^j represents the index of the j-th neighboring vertex of the vertex i as shown in Fig. 6 (a). Because the evolving surface is derived from an icosphere and the topology is maintained during evolving, the maximum number of neighboring vertices is six and EA remains unchanged. In OpenGL, we implement VA and EA as two buffer objects using the *glCreateBuffers* API. Once VA and EA are initialized by tessellating an icosahedron, VA is bound to *GL_ARRAY_BUFFER* for rendering. *glDrawArrays(GL_POINTS, 0, n)* is then called to invoke the vertex shader on all vertices. In the shader, Eq. (1)–Eq. (9) are implemented to update the position of the current vertex that is being processed. The brain volume data is accessed in the shader by 3D texture fetching. The index of the current vertex is obtained using the built-in variable *gl_VertexID* in the shader. To access the neighboring vertices of the current vertex, we have to access the contents of VA and EA in the shader. This can be achieved by attaching the two buffer objects VA and EA to two texture buffer objects and accessing the contents by texture fetching. Once the neighboring vertices are available, the unit normal vector $\boldsymbol{n}$ of the current vertex can be calculated by averaging the normals of its adjacent triangle faces. After the vertex is updated, the new position is written to a variable whose value is captured to a buffer object (namely, TA) which is bound to *GL_TRANSFORM_FEEDBACK_BUFFER*. The buffer object TA containing the updated surface geometry is then bound to *GL_ARRAY_BUFFER* for the next iteration (rendering pass) and the buffer object VA is bound to *GL_TRANSFORM_FEEDBACK_BUFFER* for receiving the updated geometry. By alternately binding VA and TA to *GL_ARRAY_BUFFER* and *GL_TRANSFORM_FEEDBACK_BUFFER*, we can avoid data transferring between the GPU and CPU thus saving the time cost. Figure 7 illustrates the shader-based OpenGL pipeline for surface evolution.

The last problem concerning computational efficiency lies in icosphere tessellation. As shown in Fig. 6 (b), in each time of tessellation, a triangle is tessellated into four triangles by adding three new vertices V_{12}, V_{13} and V_{23}. However, the new vertices may have already been generated when tessellating adjacent triangles who share the same edge. Therefore vertex duplication should be tested. Because we do not want redundant vertex data during the tessellation, a common idea is to delete duplicated vertex data after tessellation is completely done, which has the time complexity of $O(n^2)$. We propose to detect vertex duplication using a self-balancing binary search tree when adding tessellated vertices into the vertex array, which reduces the time complexity to $O(n \log_2 n)$. Assume we are tessellating a triangle V_1-V_2-V_3 represented by the vertex indices. Take the V_{12} vertex as an example, we first encode the vertex into a 64-bit integer K by packing the 32-bit integer representation of the smaller number among V_1 and V_2 into the lower 32 bit and packing the larger one into the higher 32 bit of K. Then, we search the binary tree for K. If K already exists in the tree which means V_{12} already exists, its index is retrieved to add new triangles. Otherwise K is inserted into

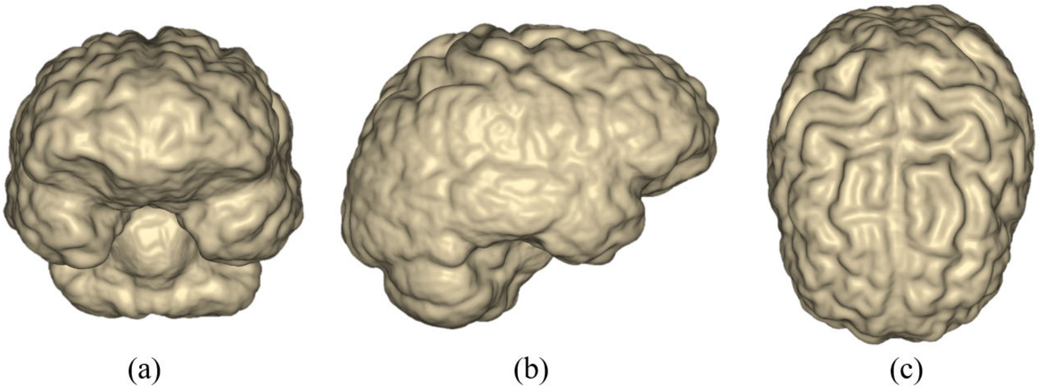


FIGURE 8. Brain surface. (a) Front view. (b) Lateral view. (c) Top view.

the tree and the coordinates of V_{12} are calculated using V_1 and V_2. At last, the coordinates of V_{12} are added into the vertex array with the resulting index being used to add new triangles. A good implementation of the self-balancing binary search tree is the Map container in the standard template library of C++. After the final tessellation, we obtain a vertex array without repetition and a triangle list defining the topology of the icosphere. The connectivity array EA is then generated from the triangle list by traversing every edge and adding the vertices of each edge to their mutual adjacent vertex lists. The adjacent vertices of a vertex are finally clockwise re-ordered.

C. MASK GENERATION

After surface evolution stops (e.g., reach the maximum number of iterations or a convergence is detected), a smooth, closed brain surface mesh is obtained as shown in Fig. 8. Next, a binary 3D mask of the original brain volume is generated according to the brain surface (brain segmentation). The voxels inside the brain surface are set to be 1 and those outside the brain surface are set to be 0. The mask is used for skull stripping and vivid volume rendering of the brain. Although the brain surface generated by the improved BET algorithm is a fine mesh model, it has weak shape resolution in the region of cerebral cortex due to the inclusion of CSF between GM inside the brain surface. By combining the binary mask and the opacity/color transfer function in the volume rendering, we can achieve high-resolution comprehensive visualization of the brain.

To generate the mask, one possible method is to iterate over every voxel of the mask and test the voxel whether inside the surface or not by counting the number of intersection points between the ray emitted from the voxel and the surface. However this method is impractical due to the high computational cost. We propose a more efficient approach by first computing 2D closed polygonal contours where the brain volume's axial slices intersect with the brain surface, and then filling the contours slice by slice.

A geometry shader is used to generate the intersection polygonal contours. Since the brain surface is a triangle mesh, in the geometry shader we calculate the intersection line (if exists) between the axial plane and the triangle of being processed. If two edges of a triangle intersect with a plane, there exists a intersection line. Assume $\boldsymbol{t_1}$-$\boldsymbol{t_2}$ are one edge of a triangle, the intersection point $\boldsymbol{p}$ with a plane passing an point $\boldsymbol{o}_p$ with the unit normal vector $\boldsymbol{n}_p$ is given by:

$$a = (\boldsymbol{o}_p - \boldsymbol{t}_1) \cdot \boldsymbol{n}_p \tag{11}$$

$$b = (\boldsymbol{o}_p - \boldsymbol{t}_2) \cdot \boldsymbol{n}_p \tag{12}$$

$$\alpha = |a|/(|a| + |b|) \tag{13}$$

$$\boldsymbol{p} = \begin{cases} (1-\alpha)\boldsymbol{t}_1 + \alpha\boldsymbol{t}_2 & \text{if } ab \leq 0 \\ \text{no intersection} & \text{otherwise} \end{cases} \tag{14}$$

Because the shader is invoked on all triangles of the brain surface, these individual intersection lines are captured back using the transform feedback mechanism of OpenGL and compose the polygonal contour. Shaders are invoked in parallel, therefore this procedure is extremely fast. The last step is to fill the polygonal contour slice by slice to generate the 3D mask. A scan line polygon fill algorithm is employed to fill the polygon. For each horizontal scan line of the slice image, calculate the intersection points with the polygonal contour and arrange the intersection points from left to right in terms of the column coordinates. The pixels between $2i-1$-th and $2i$-th ($i = 1, 2, \cdots$) intersection points are filled with 1. Figure 9 shows the mask generation result using the proposed approach.

D. VISUALIZATION

A mask based ray casting algorithm is implemented to visualize the brain volume using OpenGL. A fragment shader is used to cast a ray passing through the volume data. The ray's mathematical representation is determined by the camera model in the form of the model-view-projection matrix. For each sample point on the ray with the mask value being nonzero, its color and opacity are calculated using color/opacity transfer functions provided by the user. The sample point is further shaded using the color with a lighting model. A front-to-back composition algorithm is applied to synthesize the final color of the ray contributed by all the

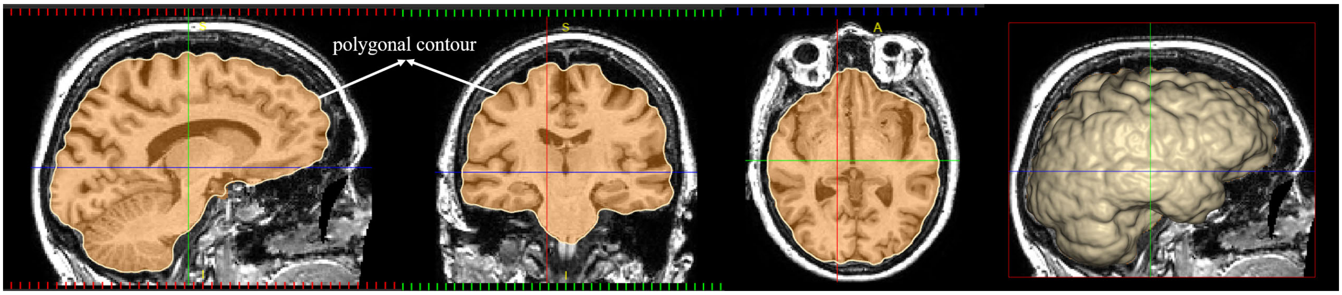


FIGURE 9. Mask generation.

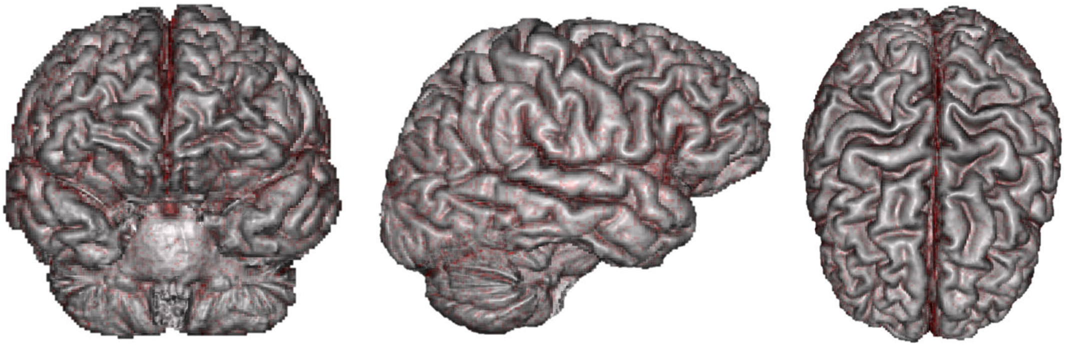


FIGURE 10. Brain visualization. Compared with Fig. 8, the sulci and gyri is clearly present.

sample points whose mask values are nonzero. The fragment shader runs in parallel on all fragments that comprises the final rendering image. Figure 10 demonstrates the brain visualization result and Figure 11 illustrates the entire processing framework for brain extraction and visualization.

For sample point shading, the normal vector of the implicit surface embedded in the 3D volume is necessary, calling for the computation of the 3D gradient images of the original brain MR volume. The 3D Sobel kernels are convolved with the volume to obtain the gradient images in x, y and z directions. However, 3D convolution is time consuming. We adopt a compute shader to perform the convolution in parallel. A work group with 16 × 16 × 4 threads is designed to simultaneously calculate the gradients within a sub region of 14 × 14 × 2. A thread is one invocation of a computer shader. A 3D grid of work groups that covers the whole brain volume is then dispatched to compute the gradient images.

IV. EXPERIMENTS AND RESULTS

A. SETUP

The proposed framework for fast brain extraction and visualization was implemented using C++ and OpenGL 4.5 APIs. We have developed a GUI desktop application[1] integrating the framework using Qt 5.11 for display and user interaction as shown in Fig. 12. The application provides data loading (supporting image formats of NifTI, DICOM, mhd, hdr, etc.), brain extraction, mask generation, interactive volume/surface rendering, and data serialization functionalities. The application was running on a workstation with an Intel Xeon E5-2609 CPU (1.90GHz) and a GeForce GTX Titan X GPU. The surface evolution process automatically stops by checking whether the maximum displacement of vertices is below a threshold of 0.25 mm. The number of tessellation times was 5 yielding an icosphere of 10242 vertices. By default, the parameters used in our method were $b_t = 0.5$, $d_2 = 20$, $\theta = 0$. The number of maximum iterations was set to be 10000. We compared our method with publicly available widely used skull stripping tools (BET in FSL v6.0,[2] BrainSuite 19a,[3] FreeSurfer V6.0,[4] AFNI V19.1.26,[5] FireVoxel (build 301)[6]) in terms of computational time and segmentation accuracy on three brain MR image datasets. Since brain extraction results are affected by parameter choice in almost all methods, in our evaluation we took the best results for comparison by fine-tuning parameters of each candidate method. The parameter tuning process of our method is simple:

- Run the algorithm using the default parameters.
- If the segmentation is entirely smaller than the real brain, decrease b_t to 0.4.

[1] https://mrs.buaa.edu.cn/?page_id=342

[2] https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/

[3] http://brainsuite.org/

[4] https://surfer.nmr.mgh.harvard.edu/

[5] https://afni.nimh.nih.gov/

[6] https://wp.nyu.edu/firevoxel/

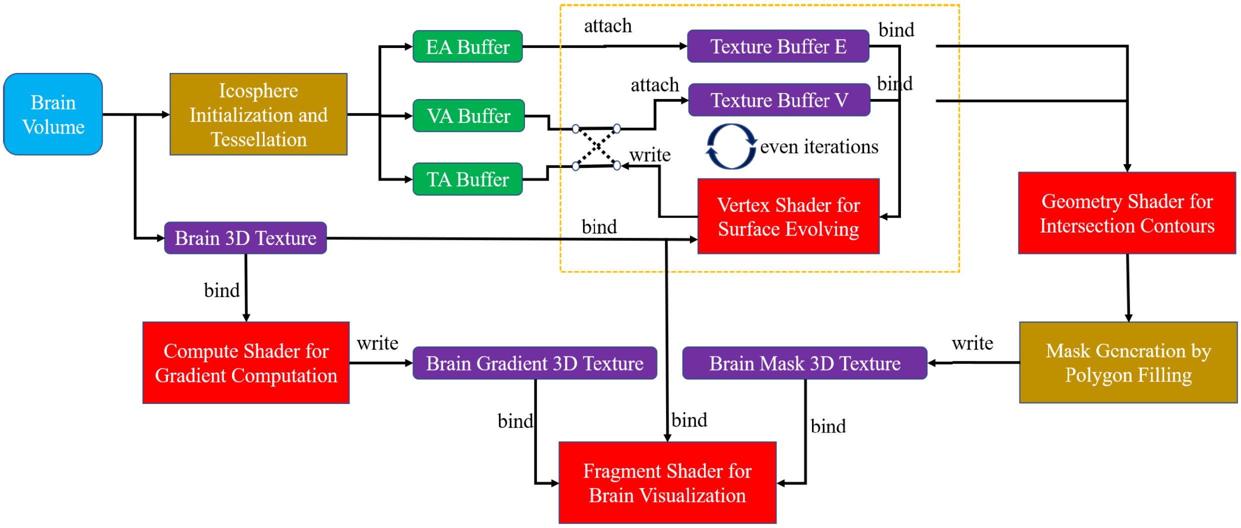

Brain Volume
Icosphere Initialization and Tessellation
EA Buffer
VA Buffer
TA Buffer
attach
attach
write
Texture Buffer E
Texture Buffer V
bind
bind
even iterations
Vertex Shader for Surface Evolving
Geometry Shader for Intersection Contours
Brain 3D Texture
bind
bind
Compute Shader for Gradient Computation
write
Brain Gradient 3D Texture
Brain Mask 3D Texture
write
Mask Generation by Polygon Filling
bind
bind
bind
Fragment Shader for Brain Visualization

FIGURE 11. Integrated framework for brain extraction and visualization.

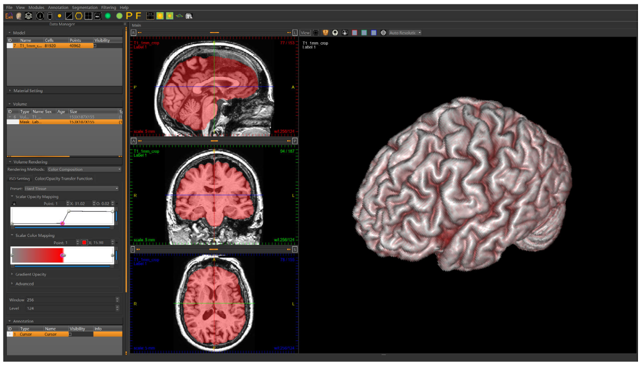


FIGURE 12. GUI desktop application for fast brain extraction and visualization.

- Check the segmentation in the region of cerebral cortex, if underestimation happens, decrease d_2 to 5 mm.
- Check the segmentation in the region of posterior meningeal septa, if overestimation/underestimation happens increase/decrease θ by a step of 0.1π.

Because the brain extraction results can be visualized instantly, the tuning process is very fast and intuitive. For other methods, the recommended parameters with trial-and-error were adopted:

- BET: start with $b_t = 0.5$ and adjust it according to the result.
- BrainSuite: automatic parameter tuning on by the program.
- FreeSurfer: (preweight the input image using atlas information)-w = 0.82, (use the basins merging atlas information)-b = 0.32, (presize the preflooding height)-h = 10,-seedpt,-ta,-wta.
- AFNI: -shrink fac bot lin = 0.65 -shrink fac = 0.72 for the IBSR dataset and -shrink fac bot lin = 0.65 -shrink fac = 0.6 for other datasets.
- FireVoxel: plane for seed search = axial, SI low = 0.528 relative to seed average, SI high = 1.35 relative to seed average, peel distance = 2.9 mm, grow distance = 6.4 mm, strict CoreSet surface = true, use edges = off, Subvoxel level = 1.

B. DATASET

- IBSR_V2.0 skull-stripped NifTI [46] obtained from the Internet Brain Segmentation Repository (IBSR)[7] contains 18 subjects, 1.5 mm T1-weighted scans in NifTI format with hand labelling brain mask as ground truth.
- NFBS skull-stripped images [47] obtained from The Neurofeedback Skull-stripped (NFBS) repository[8] is a database of 125 T1-weighted anatomical MRI scans that are manually skull-stripped. The data was collected from 125 participants, 21 to 45 years old, with a variety of clinical and subclinical psychiatric symptoms. The resolution of the images is 1 mm^3 and each file is in NiFTI format.
- Anonymous clinical data obtained from our collaborated hospital contains 8 subjects, 0.9 mm T1-weighted head scans with the series protocol of t1_mpr_sag_p2_iso in DICOM format. The average image resolution is 168x264x218. The example slices are shown in Fig. 13. Brain masks were delineated by professional neurosurgeons as the ground truth.

C. RESULTS

Dice coefficient (DC) is adopted to calculates the similarity of two masks A and B as follows:

$$DC(A, B) = \frac{2|A \cap B|}{|A| + |B|} \tag{15}$$

[7] https://www.nitrc.org/projects/ibsr
[8] http://preprocessed-connectomes-project.org

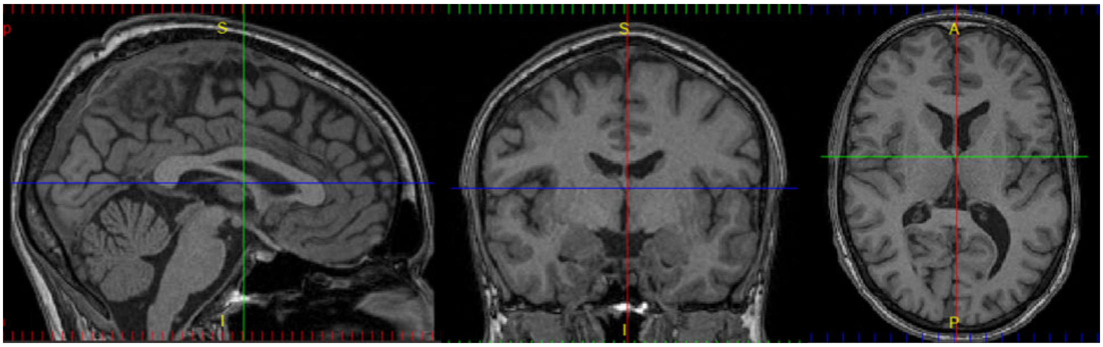


FIGURE 13. Example slices of the clinical data.

TABLE 1. Mean DICE and time cost on three datasets.

Dataset Name	BET		BrainSuite		FreeSurfer		AFNI		FireVoxel		Our Method	
	DC (%)	Time (ms)	DC	Time	DC	Time	DC	Time	DC	Time	DC	Time
IBSR	96.3	1796	95.6	11222	92.9	11817	93.8	86910	87.3	5574	**96.8**	**361**
NFBS	94.5	1586	94.0	14020	89.4	11785	92.5	76090	87.8	2642	**97.1**	**341**
Clinical Data	96.3	2289	95.2	12750	92.1	14189	93.7	65022	91.9	7116	**98.5**	**502**

TABLE 2. Surface metrics on three datasets (unit: mm).

Methods	IBSR				NFBS				Clinical Data			
	HD	ASSD	SDD	MSSD	HD	ASSD	SDD	MSSD	HD	ASSD	SDD	MSSD
Our Method	8.51	0.92	1.11	12.44	10.44	0.79	1.15	12.58	7.28	0.40	0.89	12.93
BET	14.23	1.13	1.82	14.72	14.58	1.66	1.75	15.29	16.48	1.20	1.80	17.80
BrainSuite	15.08	1.44	1.96	16.42	14.62	1.93	1.90	15.87	15.46	1.67	1.94	16.40
FreeSurfer	15.10	2.65	2.66	20.72	16.90	3.58	2.37	35.60	16.83	2.72	2.24	22.85
AFNI	12.69	2.16	2.05	14.92	11.90	2.48	1.58	13.24	10.33	2.11	1.41	14.57
FireVoxel	29.57	5.41	7.66	45.69	13.44	6.59	8.06	42.12	19.25	4.96	8.09	43.67

We calculate the Dice coefficient between the ground truth and the output mask of each method to evaluate the accuracy of the brain extraction. In addition, the running time of each method was also recorded. Our method achieved highest mean Dice coefficients of 96.8%, 97.1%, 98.5% and lowest mean time cost of 361 ms, 338 ms, 502 ms on the three datasets, outperforming all the other methods (with p-values less than 0.01). The mean Dice coefficients and time cost of each method on the three datasets are listed in Table 1. The statistical results are graphically represented in Fig. 14 using the box plot. Note that the time axis is logarithmic. In addition, our method provides instant comprehensive brain visualization by virtue of the integrated OpenGL pipeline, yielding a rendering speed of more than 60 frames per second (fps).

In addition to voxel overlap evaluation, we have also calculated surface metrics of Hausdorff distance (HD), average symmetric surface distance (ASSD), maximum symmetric surface distance (MSSD), surface distance deviation (SDD) to evaluate geometric differences between the ground truth and the segmented brain. The results on the three datasets (average over the whole dataset) are shown in Table 2. Our method achieved the optimal metrics on all the datasets.

The secondly well performed algorithm was BET. Our method defeated the original BET in those cases where a large amount of CSF exists between GM in the region of cerebral cortex (deep sulci and gyri). A typical case is shown in Fig. 15(a) from a patient suffering from encephalatrophy. Figure. 15(b) shows the best segmentation result by BET with $b_t = 0.2$. Segmentation errors are observed in the region of sulci and gyri. In contrast, our method gives better segmentation result as shown in Fig. 15(c) with $b_t = 0.4$, $d_2 = 5$, $\theta = -0.4\pi$. Figure. 15(d) shows the instant volume rendering result of the skull-stripped brain.

V. DISCUSSION

Since every subject has common statistical characteristics with individual variations, there is no such a gold segmentation algorithm that can generalize well for a large scale of datasets. For a specific algorithm, there is also a tradeoff between the number of parameters available for tuning and fineness of segmentation. To improve the segmentation fineness, you may have to open more parameters for tuning, this however calls for more patience and tricks, leading to increased time cost and labors which may prevent the prevalence of the tool in clinical practice. Our method introduces two additional parameters d_2 and θ for tuning compared with

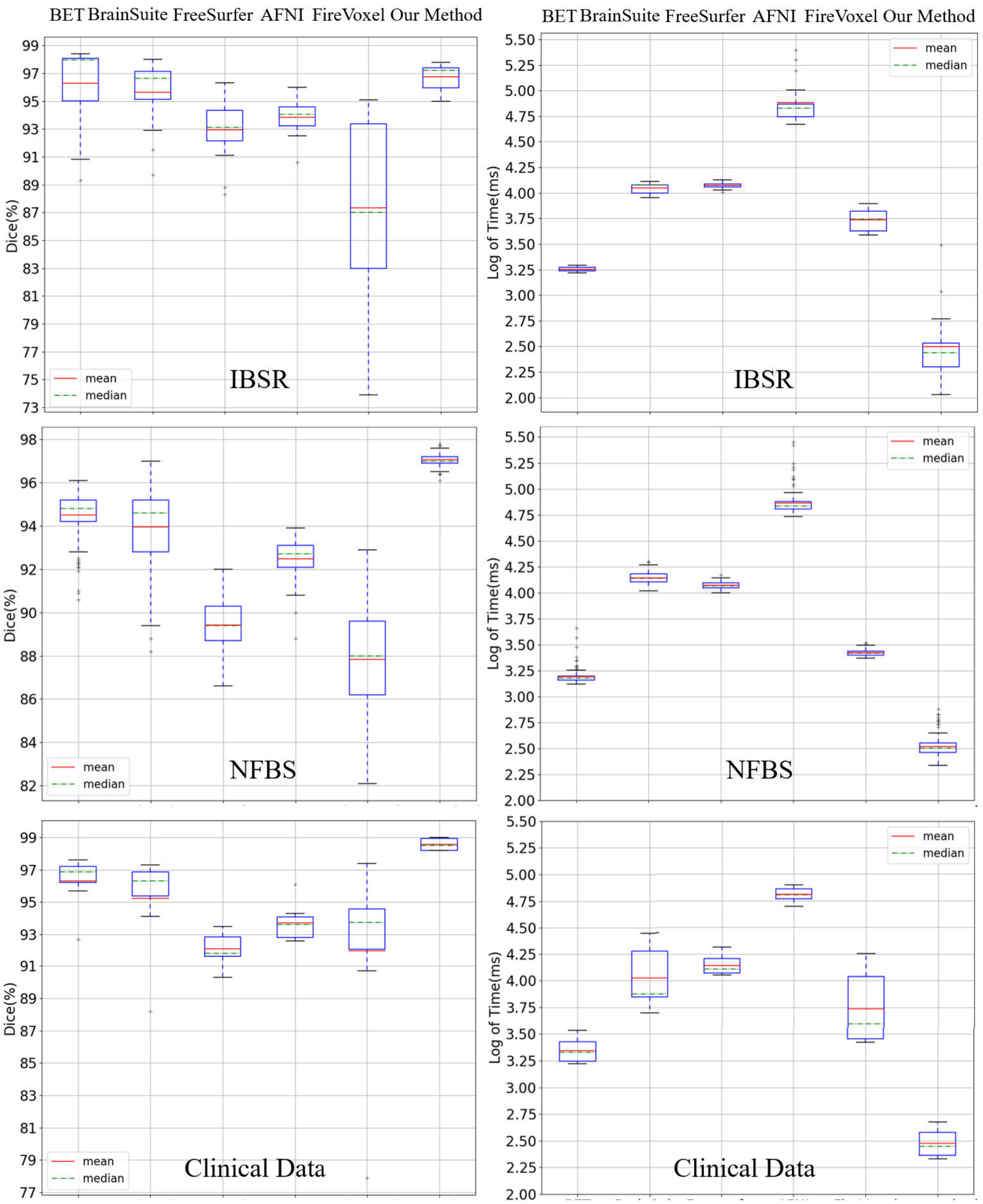


FIGURE 14. Performance of our methods compared with other methods.

the original BET to achieve good segmentation in the region of cerebral cortex. In most cases d_2 has only three choices: 20, 10 or 5. θ can be adjusted with a step of 0.1π from $-\pi/2$ to $\pi/2$. Because we have instant visualization of the segmentation result, these parameters can be easily and quickly determined by visual inspection. The brain extraction processing framework is implemented using modern OpenGL APIs and is seamlessly integrated with the rendering pipeline. The whole framework achieves an end-to-end functionality from MR image input to comprehensive brain visualization within 1-2 seconds by virtue of a GPU. Because the instant brain visualization is provided, the parameter tuning becomes simple and intuitive and can be finished in several seconds. Our method is very suitable for intraoperative image guided neurosurgery, because the time lag between the image acquisition and accurate comprehensive brain visualization is within several seconds.

The methodology innovation of our method compared with the original BET lies in the concept of d-field. The image force f_3 is determined by the local intensity profile up to the depth d. As illustrated in Fig. 3, the choice of d affects the direction of the image force. The larger d is, the more

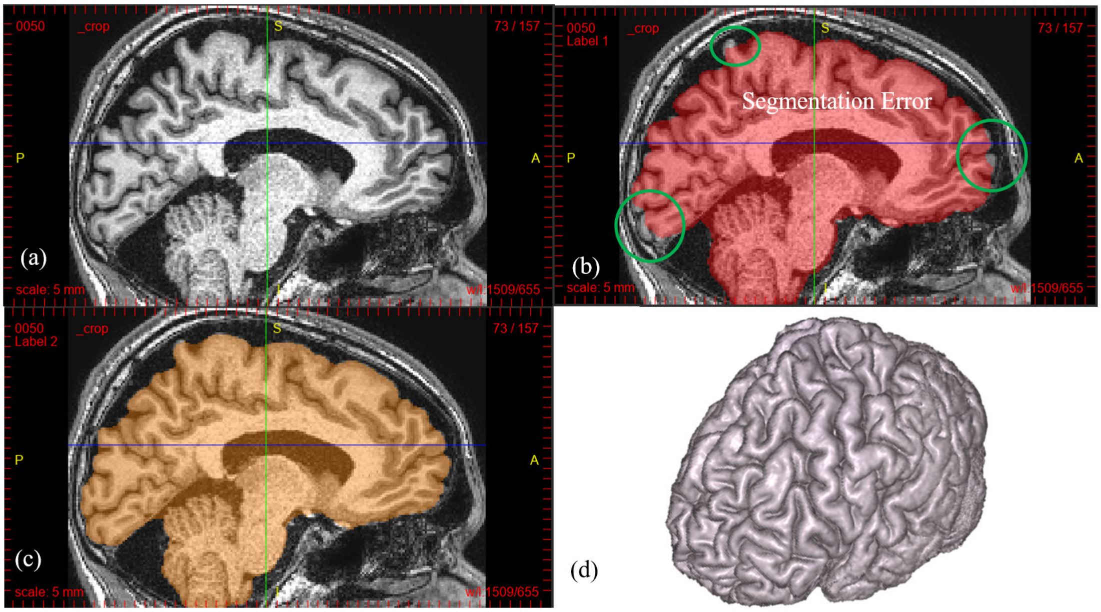


FIGURE 15. MR image of encephalatrophy. (a) MR images show the deep sulci and gyri. (b) BET extraction results with segmentation errors observed in the superficial cortical layers ($b_t = 0.2$). (c) Segmentation results using our method ($b_t = 0.4$, $d_2 = 5$, $\theta = -0.4\pi$) with improved segmentation accuracy. (d) Instant brain visualization provided by our method.

damping effect of the surface expansion. In the view of machine learning, d serves as an important local feature to discriminate brain and non-brain tissues. In the region of superficial cortical layers with deep sulci and gyri, a smaller d is desired to drive the surface to penetrate CSF between GM, while in the other regions a larger d is needed for robust convergence of the evolving surface to the real brain surface. The constant d apparently has not enough discrimination power, and the tuning of b_t only leads to entire expansion or contraction. By using the r (curvature)-d (searching depth) mapping with region division, we create a varying field of d that has good discrimination power for fine segmentation. The purpose of region division is to identify the brain part where a smaller d_2 is applied to avoid underestimation of superficial cortical layers. The current setting includes the part of upper cerebral cortex and θ is used to possibly exclude the posterior meningeal septa as shown in Fig. 5(b) for some cases where the brain tissue is squeezing against the posterior meningeal septa.

VI. CONCLUSION

In this paper we propose and implement a fast brain extraction and visualization framework based on modern OpenGL pipelines which can fully exploit the parallel computing capability of a GPU. A 3D surface evolution algorithm based on the BET algorithm is presented with the innovation of d-field to improve the segmentation accuracy in the region of cerebral cortex with deep sulci and gyri. The proposed framework achieves an end-to-end functionality from brain MR image input to comprehensive brain visualization within 1-2 seconds and yields a rendering frame rate of more than 60 fps. Experiments were performed on two publicly available datasets and one clinical dataset to compare our method with five state-of-the-art skull stripping tools including the original BET in terms of segmentation accuracy and time cost. The proposed method is supposed to fill the gap between intraoperative image acquisition and accurate comprehensive 3D visualization in neurosurgical navigation.

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

REFERENCES

[1] I. Despotović, B. Goossens, and W. Philips, "MRI segmentation of the human brain: Challenges, methods, and applications," *Comput. Math. Methods Med.*, vol. 2015, pp. 1–23, Oct. 2015.

[2] S. Bauer, C. May, D. Dionysiou, G. Stamatakos, P. Buchler, and M. Reyes, "Multiscale modeling for image analysis of brain tumor studies," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 1, pp. 25–29, Jan. 2012.

[3] M. Shakarami, A. A. Suratgar, and H. A. Talebi, "Intraoperative brain shift estimation using atlas of brain deformations and constrained Kalman filter," *IEEE Trans. Control Syst. Technol.*, to be published.

[4] S. Bauer, R. Wiest, L.-P. Nolte, and M. Reyes, "A survey of MRI-based medical image analysis for brain tumor studies," *Phys. Med. Biol.*, vol. 58, no. 13, pp. R97–R129, 2013.

[5] P. Coupé, S. F. Eskildsen, J. V. Manjón, V. S. Fonov, and D. L. Collins, "Simultaneous segmentation and grading of anatomical structures for patient's classification: Application to Alzheimer's disease," *NeuroImage*, vol. 59, no. 4, pp. 3736–3747, 2012.

[6] N. Archip, O. Clatz, S. Whalen, D. Kacher, A. Fedorov, A. Kot, N. Chrisochoides, F. Jolesz, A. Golby, P. M. Black, and S. K. Warfield, "Non-rigid alignment of pre-operative MRI, fMRI, and DT-MRI with intra-operative MRI for enhanced visualization and navigation in image-guided neurosurgery," *NeuroImage*, vol. 35, no. 2, pp. 609–624, 2007.

[7] H. Liao, N. Hata, S. Nakajima, M. Iwahara, I. Sakuma, and T. Dohi, "Surgical navigation by autostereoscopic image overlay of integral videography," *IEEE Trans. Inf. Technol. Biomed.*, vol. 8, no. 2, pp. 114–121, Jun. 2004.

[8] H. Liao, T. Inomata, I. Sakuma, and T. Dohi, "3-D augmented reality for MRI-guided surgery using integral videography autostereoscopic image overlay," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 6, pp. 1476–1486, Jun. 2010.

[9] O. Clatz, H. Delingette, I.-F. Talos, A. J. Golby, R. Kikinis, F. A. Jolesz, N. Ayache, and S. K. Warfield, "Robust nonrigid registration to capture brain shift from intraoperative MRI," *IEEE Trans. Med. Imag.*, vol. 24, no. 11, pp. 1417–1427, Nov. 2005.

[10] P. L. Kubben, K. J. ter Meulen, O. E. Schijns, M. P. ter Laak-Poort, J. J. van Overbeeke, and H. van Santbrink, ''Intraoperative MRI-guided resection of glioblastoma multiforme: A systematic review,'' *The Lancet Oncol.*, vol. 12, no. 11, pp. 1062–1070, 2011.

[11] P. Fatemi, M. Zhang, K. J. Miller, P. Robe, and G. Li, ''How intraoperative tools and techniques have changed the approach to brain tumor surgery,'' *Current Oncol. Rep.*, vol. 20, no. 11, p. 89, Sep. 2018.

[12] A. G. Balan, A. J. Traina, M. X. Ribeiro, P. M. Marques, and C. Traina, Jr., ''Smart histogram analysis applied to the skull-stripping problem in T1-weighted MRI,'' *Comput. Biol. Med.*, vol. 42, no. 5, pp. 509–522, 2012.

[13] A. Subudhi, J. Jena, and S. Sabut, ''Extraction of brain from MRI images by skull stripping using histogram partitioning with maximum entropy divergence,'' in *Proc. Int. Conf. Commun. Signal Process. (ICCSP)*, Apr. 2016, pp. 0931–0935.

[14] H. Khastavaneh and H. Ebrahimpour-Komleh, ''Brain extraction: A region based histogram analysis strategy,'' in *Proc. Signal Process. Intell. Syst. Conf. (SPIS)*, Dec. 2015, pp. 20–24.

[15] F. J. Galdames, F. Jaillet, and C. A. Perez, ''An accurate skull stripping method based on simplex meshes and histogram analysis for magnetic resonance images,'' *J. Neurosci. Methods*, vol. 206, no. 2, pp. 103–119, 2012.

[16] A. Mikheev, G. Nevsky, S. Govindan, R. Grossman, and H. Rusinek, ''Fully automatic segmentation of the brain from T1-weighted MRI using *Bridge Burner* algorithm,'' *J. Magn. Reson. Imag.*, vol. 27, no. 6, pp. 1235–1241, 2008.

[17] L. Lemieux, G. Hagemann, K. Krakow, and F. G. Woermann, ''Fast, accurate, and reproducible automatic segmentation of the brain in T_1-weighted volume MRI data,'' *Magn. Reson. Med.*, vol. 42, no. 1, pp. 127–135, 1999.

[18] M. E. Brummer, R. M. Mersereau, R. L. Eisner, and R. R. J. Lewine, ''Automatic detection of brain contours in MRI data sets,'' *IEEE Trans. Med. Imag.*, vol. 12, no. 2, pp. 153–166, Jun. 1993.

[19] D. W. Shattuck, S. R. Sandor-Leahy, K. A. Schaper, D. A. Rottenberg, and R. M. Leahy, ''Magnetic resonance image tissue classification using a partial vol. model,'' *NeuroImage*, vol. 13, no. 5, pp. 856–876, 2001.

[20] S. Sandor and R. Leahy, ''Surface-based labeling of cortical anatomy using a deformable atlas,'' *IEEE Trans. Med. Imag.*, vol. 16, no. 1, pp. 41–54, Feb. 1997.

[21] Y. Zhou and J. Bai, ''Atlas-based fuzzy connectedness segmentation and intensity nonuniformity correction applied to brain MRI,'' *IEEE Trans. Biomed. Eng.*, vol. 54, no. 1, pp. 122–129, Jan. 2007.

[22] K. K. Leung, J. Barnes, M. Modat, G. R. Ridgway, J. W. Bartlett, N. C. Fox, S. Ourselin, and Alzheimer's Disease Neuroimaging Initiative, ''Brain MAPS: An automated, accurate and robust brain extraction technique using a template library,'' *NeuroImage*, vol. 55, no. 3, pp. 1091–1108, 2011.

[23] C. Ledig, R. A. Heckemann, A. Hammers, J. C. Lopez, V. F. J. Newcombe, A. Makropoulos, J. Lötjönen, D. K. Menon, and D. Rueckert, ''Robust whole-brain segmentation: Application to traumatic brain injury,'' *Med. Image Anal.*, vol. 21, no. 1, pp. 40–58, 2015.

[24] S. F. Eskildsen, P. Coupé, V. Fonov, J. V. Manjón, K. K. Leung, N. Guizard, S. N. Wassef, L. R. Østergaard, and D. L. Collins, ''BEaST: Brain extraction based on nonlocal segmentation technique,'' *NeuroImage*, vol. 59, no. 3, pp. 2362–2373, 2012.

[25] M. S. Atkins and B. T. Mackiewich, ''Fully automatic segmentation of the brain in MRI,'' *IEEE Trans. Med. Imag.*, vol. 17, no. 1, pp. 98–107, Feb. 1998.

[26] M. E. Leventon, W. E. L. Grimson, and O. Faugeras, ''Statistical shape influence in geodesic active contours,'' in *Proc. 5th IEEE EMBS Int. Summer School Biomed. Imag.*, Jun. 2002, pp. 1–8.

[27] H.-T. Liu, T. W. H. Sheu, and H.-H. Chang, ''Automatic segmentation of brain mr images using an adaptive balloon snake model with fuzzy classification,'' *Med. Biol. Eng. Comput.*, vol. 51, no. 10, pp. 1091–1104, Oct. 2013.

[28] S. M. Smith, ''Fast robust automated brain extraction,'' *Hum. Brain Mapping*, vol. 17, no. 3, pp. 143–155, 2002.

[29] A. H. Zhuang, D. J. Valentino, and A. W. Toga, ''Skull-stripping magnetic resonance brain images using a model-based level set,'' *NeuroImage*, vol. 32, no. 1, pp. 79–92, 2006.

[30] M. Battaglini, S. M. Smith, S. Brogi, and N. D. Stefano, ''Enhanced brain extraction improves the accuracy of brain atrophy estimation,'' *NeuroImage*, vol. 40, no. 2, pp. 583–589, 2008.

[31] Y. Zhang, M. Brady, and S. Smith, ''Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm,'' *IEEE Trans. Med. Imag.*, vol. 20, no. 1, pp. 45–57, Jan. 2001.

[32] F. Shi, L. Wang, Y. Dai, J. H. Gilmore, W. Lin, and D. Shen, ''LABEL: Pediatric brain extraction using learning-based meta-algorithm,'' *NeuroImage*, vol. 62, no. 3, pp. 1975–1986, 2012.

[33] J. Kleesiek, G. Urban, A. Hubert, D. Schwarz, K. Maier-Hein, M. Bendszus, and A. Biller, ''Deep MRI brain extraction: A 3D convolutional neural network for skull stripping,'' *NeuroImage*, vol. 129, pp. 460–469, Apr. 2016.

[34] P. Moeskops, M. A. Viergever, A. M. Mendrik, L. S. de Vries, M. J. N. L. Benders, and I. Išgum, ''Automatic segmentation of MR brain images with a convolutional neural network,'' *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1252–1261, May 2016.

[35] H. Hwang, H. Z. U. Rehman, and S. Lee, ''3D U-net for skull stripping in brain MRI,'' *Appl. Sci.*, vol. 9, no. 3, p. 569, 2019.

[36] F. Ségonne, A. Dale, E. Busa, M. Glessner, D. Salat, H. Hahn, and B. Fischl, ''A hybrid approach to the skull stripping problem in MRI,'' *NeuroImage*, vol. 22, no. 3, pp. 1060–1075, 2004.

[37] C. Ciofolo and C. Barillot, ''Atlas-based segmentation of 3D cerebral structures with competitive level sets and fuzzy control,'' *Med. Image Anal.*, vol. 13, no. 3, pp. 456–470, 2009.

[38] C. Wang and Ö. Smedby, ''Multi-organ segmentation using shape model guided local phase analysis,'' in *Medical Image Computing and Computer-Assisted Intervention—MICCAI*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham, Switzerland: Springer, 2015, pp. 149–156.

[39] S. A. Sadananthan, W. Zheng, M. W. Chee, and V. Zagorodnov, ''Skull stripping using graph cuts,'' *NeuroImage*, vol. 49, no. 1, pp. 225–239, 2010.

[40] G. Wu, Q. Wang, D. Zhang, F. Nie, H. Huang, and D. Shen, ''A generative probability model of joint label fusion for multi-atlas based brain segmentation,'' *Med. Image Anal.*, vol. 18, no. 6, pp. 881–890, 2014.

[41] M. Kass, A. Witkin, and D. Terzopoulos, ''Snakes: Active contour models,'' *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, 1988.

[42] V. Caselles, R. Kimmel, and G. Sapiro, ''Geodesic active contours,'' *Int. J. Comput. Vis.*, vol. 22, no. 1, pp. 61–79, Feb. 1997.

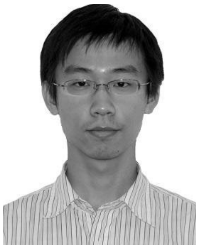
[43] J. E. Iglesias, C. Y. Liu, P. M. Thompson, and Z. Tu, ''Robust brain extraction across datasets and comparison with publicly available methods,'' *IEEE Trans. Med. Imag.*, vol. 30, no. 9, pp. 1617–1634, Sep. 2011.

[44] Ø. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, ''3D U-net: Learning dense volumetric segmentation from sparse annotation,'' in *Medical Image Computing and Computer-Assisted Intervention—MICCAI*, S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal, and W. Wells, Eds. Cham: Springer, 2016, pp. 424–432.

[45] F. Fischer, M. A. Selver, W. Hillen, and C. Guzelis, ''Integrating segmentation methods from different tools into a visualization program using an object-based plug-in interface,'' *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 4, pp. 923–934, Jul. 2010.

[46] T. Rohlfing, ''Image similarity and tissue overlaps as surrogates for image registration accuracy: Widely used but unreliable,'' *IEEE Trans. Med. Imag.*, vol. 31, no. 2, pp. 153–163, Feb. 2012.

[47] B. Puccio, J. P. Pooley, J. S. Pellman, E. C. Taverna, and R. C. Craddock, ''The preprocessed connectomes project repository of manually corrected skull-stripped T1-weighted anatomical MRI data,'' *Gigascience*, vol. 5, no. 45, pp. 1–7, Oct. 2016.



JUNCHEN WANG received the B.S. and Ph.D. degrees in mechanical engineering from Beihang University, Beijing, China, in 2006 and 2012, respectively.

He was a Postdoctoral Fellow with the University of Tokyo, Tokyo, Japan, from 2012 to 2016. He is currently an Associate Professor with Beihang University. He has authored more than 50 peer-reviewed articles published in international journals and conference proceedings. His research interests include medical robotics, surgical navigation, medical image computing, and the fusion of these techniques for minimally invasive advanced diagnosis and therapy in clinical practice.

ZHEN SUN received the B.S. degree in mechanical engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, Sichuan, China, in 2010, and the M.S. degree in mechanical engineering from Beihang University, Beijing, China, in 2017, where he is currently pursuing the Ph.D. degree in mechanical engineering.

His research interests include medical robotics, image-guided surgery, master-slave control, and miniature robot design.

HONGLEI JI received the B.S. degree in mathematics from Beihang University, Beijing, China, in 2018, where she is currently pursuing the master's degree in mechanical engineering.

Her research interests include medical image computing and computer vision.

XIAOHUI ZHANG received the B.S. and M.S. degrees in mechanical engineering from Qingdao University, Shandong, China, in 2013 and 2016, respectively. She is currently pursuing the Ph.D. degree with Beihang University, Beijing, China.

Her research interests include surgical navigation and medical image processing.

TIANMIAO WANG received the B.S. degree in computer science from Xi'an Jiaotong University, Xi'an, China, in 1982, and the M.S. and Ph.D. degrees in computer science from Northwestern Polytechnical University, Xi'an, in 1985 and 1990, respectively.

He was a Postdoctoral Researcher with the State Key Laboratory of Intelligent Technology and Systems, Tsinghua University, Beijing, China, from 1992 to 1995. He is currently a Full Professor with the School of Mechanical Engineering and Automation, Beihang University, Beijing. He is also a Cheung Kong Scholar appointed by the Ministry of Education of China. He is a member of the Academic Degree Commission of the State Council of China.

YU SHEN received the B.S., M.S., and Ph.D. degrees from Northwest University, Xi'an, in 2006, 2009, and 2018, respectively.

She is currently a Postdoctoral Researcher with the School of Mechanical Engineering and Automation, Beihang University, Beijing, China. Her research interests include medical image computing and biosensors.