3D Interactive Segmentation with Semi-Implicit Representation and Active Learning

Jingjing Deng and Xianghua Xie

Abstract

Segmenting complex 3D geometry is a challenging task due to rich structural details and complex appearance variations of target object. Shape representation and foreground-background delineation are two of the core components of segmentation. Explicit shape models, such as mesh based representations, suffer from poor handling of topological changes. On the other hand, implicit shape models, such as level-set based representations, have limited capacity for interactive manipulation. Fully automatic segmentation for separating foreground objects from background generally utilizes non-interoperable machine learning methods, which heavily rely on the off-line training dataset and are limited to the discrimination power of the chosen model. To address these issues, we propose a novel semi-implicit representation method, namely Non-Uniform Implicit B-spline Surface (NU-IBS), which adaptively distributes parametrically blended patches according to geometrical complexity. Then, a two-stage cascade classifier is introduced to carry out efficient foreground and background delineation, where a simplistic Naïve-Bayesian model is trained for fast background elimination, followed by a stronger pseudo-3D Convolutional Neural Network (CNN) multi-scale classifier to precisely identify the foreground objects. A localized interactive and adaptive segmentation scheme is incorporated to boost the delineation accuracy by utilizing the information iteratively gained from user intervention. The segmentation result is obtained via deforming an NU-IBS according to the probabilistic interpretation of delineated regions, which also imposes a homogeneity constrain for individual segments. The proposed method is evaluated on a 3D cardiovascular Computed Tomography Angiography (CTA) image dataset and Brain Tumor Image Segmentation Benchmark 2015 (BraTS2015) 3D Magnetic Resonance Imaging (MRI) dataset.

1 Introduction

Interactive image segmentation plays an important role in computer vision, graphics, and medical image analysis, where user intervention is often used as an additional source of information to guide or refine the segmentation process, e.g. [1]. The interactive scheme that captures expert knowledge on-the-fly through user interventions can be more efficient and adaptive than fully automated methods that heavily rely on off-line training. Many traditional methods such as active contours and region growing methods are capable of incorporating basic user interactions. User interventions however can be in many diverse ways, e.g. initialization, foreground-background indication, parameter tuning, and result refinement. Interactive segmentation incorporated with statistical models has proved to be an effective strategy to differentiate foreground and background, where the supervised intervention from user is directly applied to the image, e.g. [2]. User interaction can often be considered as supervised selective labeling, e.g. [3, 4, 5, 6]. Segmentation can then be formulated as solving an either discrete or continuous optimization problem and user interaction is used as regularization constraints [7, 8]. When dealing with higher dimensional data, e.g. 3D volumetric images, computational efficiency and versatile user interaction are particularly important. We consider the following as the fundamental challenges in interactive segmentation: accurate delineation of foreground from background, effective interaction scheme supported by flexible shape representation and regularization.

In this paper, we propose a novel interactive method for volumetric image segmentation that includes state-of-the-art deep learning based object detection and novel parametric implicit shape representation (see Fig. 1). The contributions of our method can be summarized as fourfold. Firstly, the proposed NU-IBS. embeds shape into zero manifold of a level set function that is approximated using locally supported B-spline patches in parametric form. The geometrical complexity is estimated using spectral analysis and control knots are placed accordingly. It adapts to local topology and geometry, where regions with high curvature are blended using more compact patches to avoid over-smoothing. Secondly, a cascade classifier is proposed where an intensity-based Naïve-Bayesian classifier is used for fast background elimination, and a Pseudo-3D CNN classifier is followed for precise classification, which is more computationally efficient while maintaining high accuracy. The discriminative features are automatically learned in a supervised fashion together with the classification boundary. Thirdly, adaptive learning and localized refining are introduced to ensure accurate segmentation with on-the-fly user intervention. Minimal user input is required to provide foreground and background strokes which are used to adaptively fine-tune the classifier and localize the foreground. The method is able to cope with geometrical and appearance variations, as well as avoiding outliers contaminating the trained model. Finally, the piecewise constant constraint of neighborhood voxels is

Jingjing Deng and Xianghua Xie were with the Department of Computer Science, Swansea University, Bay Campus, Swansea, SA1 8EN, United Kingdom (http://csvision.swansea.ac.uk). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPU used for this research.

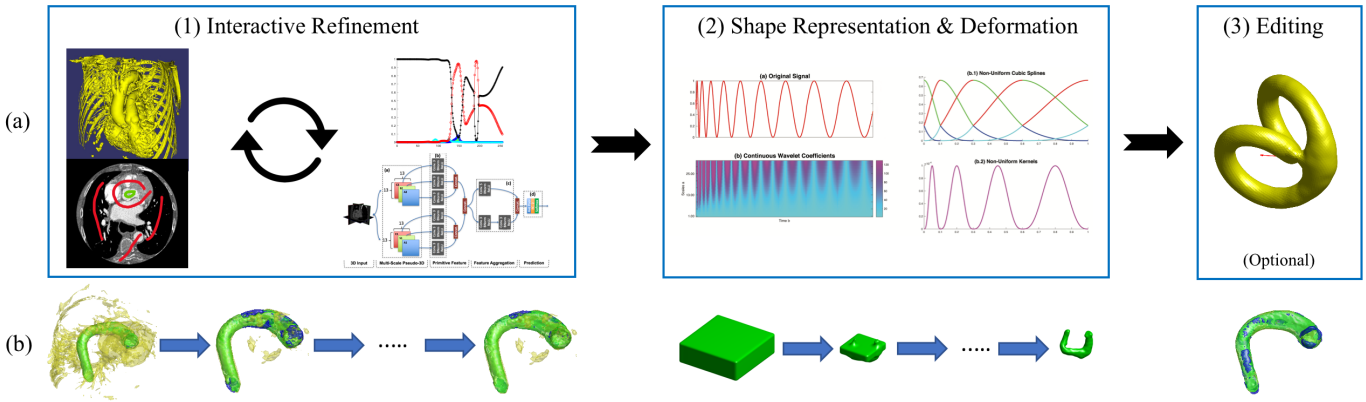


Figure 1: The flowchart of proposed interactive segmentation method. Row (a) shows the three stages of interactive segmentation. Row (b) illustrates the results from the different stages given one example case in 3D CTA dataset. At the interactive refinement stage (1), a voxel-wise classifier is used to differentiate background and foreground. The classifier consists of a Naïve Bayesian classifier and several Pseudo-3D CNN classifiers that are structured in a cascade fashion, where a CNN classifier is trained and attached to the end of the cascade pipeline where user provides a set of supervision strokes. At the shape representation and deformation stage (2), a 3D surface is constructed using the proposed NU-IBS representation and the surface propagates according to the data support given by the cascade classifier at stage (1). At the editing stage (3), user can edit the shape if it is necessary by selecting a control point and pulling the surface towards the desired direction. Stage (3) is optional and is only required when the results obtained at stage (2) is not ideal due to *i.e.* insufficient data support.

imposed to the voxel-wise classification results, where NU-IBS representation ensures geometrical smoothness, and Chan-Vese energy functional ensures regional homogeneity of separated segments. These two objectives are co-optimiszed in the total variation framework. The implicit surface propagates, according to the data support, to privide an optimal solution.

The rest of paper is organized as follows: Reviews on semi-automatic segmentation and shape representation are provided in Sec. 2. The proposed method is introduced in Sec. 3, and the experimental evaluations are presented and discussed in Sec. 4. The concluding remarks are given in Sec. 5.

2 Related Work

2.1 Foreground Delineation and User Interaction

Table 1: Related Interactive Image Segmentation Methods (F-B: Foreground-Background, ROI: Region Of Interest, GMM: Gaussian Mixture Model, CRF: Conditional Random Fields, ACP: Adaptive Constraint Propagation, FCN: Fully Convolutional Network.)

Method	Feature	Delineating Technique	Constrain Imposing	Interactive Scheme
Boykov *et al.* [9]	Grey Texture	Histogram	Graph-Cut	F-B Stroke
Rother *et al.* (GrabCut) [3]	Color Texture	GMMs	Graph-Cut	ROI Box, Border Matting
Li *et al.* (Lazy Snapping) [10]	Color Texture, Image Gradient	K-Means	Graph-Cut	F-B Stroke, Boundary Editing
Unger *el al.* (TVSeg) [5]	Color Texture	Histogram	Total Variation	ROI Box, Border Matting
Han *et al.* [11]	Color Texture, MSNST	GMMs	Graph-Cut	ROI Box
Santner *et al.* [6]	Color Texture	Random Forests	Total Variation	F-B Stroke
Gulshan *et al.* [12]	Color Texture	Random Forests	Graph-Cut	F-B Stroke
Price *et al.* (GeodesicGC) [13]	Color Texture, Geodesic Distance	Gauss. Naïve Bayesian	Graph-Cut	F-B Stroke
Meena *et al.* [14, 15]	Color Texture	Spline-based Classifier	Geometric Smoothness	Labeled Seed Points
Dong *et al.* [16]	Color Texture	Mean-Shift, GMM	Total Variation	F-B Stroke
Feng *et al.* [17]	Geodesic Distance	Naïve Bayesian	Graph-Cut	F-B Stroke
Lu *et al.* [18]	Image Gradient	Implicit, Optimization	Total Variation	Terminal Point
Isack *et al.* [19]	Color Texture	GMMs	Graph-Cut	Multi-Label Stroke
Jian *et al.* [7]	Color Texture	Mean-Shift	ACP-Cut	F-B Stroke
Xu *et al.* [20]	CNN Features	FCN	Graph-Cut	F-B Click
Wang *et al.* [21]	CNN Features	CNN	CRF-Net	F-B Stroke
Proposed Method	**Texture, CNN Features**	**Naïve Bayesian, CNN**	**Total Variation**	**F-B Stroke, Local Refining**

Semi-automatic segmentation methods typically rely on user interaction to initiate the segmentation process and some further utilize user interaction to refine the process. Interactive segmentation with graph cut is a typical example of such technique. For instance, Yuri Boykov *et al.* [9] proposed an interactive graph cut technique that can be extended to arbitrary dimensions. User guided strokes are required to label the foreground and background, and statistical distributions of these two regions are used to solve a discrete energy minimization using the max flow min-cut theorem, where a global optimal

Table 2: Parametric Implicit Representation for Surface Reconstruction and Image Segmentation

Param.	Support	Method
RBF	Global	Thin-Plate [25], Multi-Quad [26], Gauss [27]
	Local	Wendland's RBF [28]
Poly.	Global	Polynomial [29]
	Local	Uniform B-splines [30, 31, 32]
	Local	**Proposed NU-IBS**

solution exists for binary label problems. The segmentation is equivalent to a binary labeling problem given the data support and local smoothness constraint.

There are multitude variations based on graph cut method. For instance, Rother *et al.* proposed a so-called *GrabCut* method [3] for foreground and background segmentation of 2D images in RGB color space. Gaussian Mixture Models (GMMs) is used for separating foreground and background given a set of labeled bounding boxes initialized by user. Later, Gulshan *et al.* [12] demonstrated geodesic star convexity constraint can be embedded into such a framework. Han *et al.* [11] extended the *GrabCut* method by introducing multi-scale non-linear structure tensor texture feature to overcome the difficulty of delineating textured image with large scale differences. Similarly, Isack *et al.* [19] introduced hedgehog shape priors to the *GrabCut* framework for multi-object segmentation. *Lazy Snapping* [10] builds two K-Means models to partition the image into many small pre-segmented regions based on the color similarity. Therefore, the segmentation can be obtained by formulating as a binary labeling problem for pre-segmented blocks using graph cut. Unger *et al.* [5] showed that such interactive framework can be incorporated with total variation regularization, which solves an energy minimization for a geodesic active contour model. Meena *et al.* [14, 15] proposed to use labeled seed points, instead of continuous strokes, and their method showed improvements compared to the graph cut method [9].

These interactive segmentation methods fall into the same framework with a series of iterative processes as follows: acquiring foreground and background visual cues from user, delineating foreground, and then refining segmentation regularization. In order to improve efficiency, stronger delineating models with discriminative features have been proposed, such as Naïve Bayesian classifier with geodesic distance features [13] and Random Forests (RF) with a larger set of hand-crafted features [6]. Very recently, Feng *et al.* [17] showed the feasibility of applying such interactive segmentation method to RGBD images.

It is worth noting that existing methods predominately rely on statistical models to differentiate foreground from background based on user interactions. These user inputs are generally simplistic and can be biased due to, for instance, randomness in user interaction, that can lead to failure in learning. Deep learning methods have become more and more mainstream, e.g. [22, 23], and it has shown superior over many traditional methods for various visual processing tasks. In medical image segmentation, Lai [24] provided an overview on deep learning methods. However, training a deep model requires a large amount of supervision data that is generally not applicable in the case of interactive segmentation. Novel approaches are thus required in order to incorporate deep learning into interactive segmentation. In this work, we propose a two-stage cascade classifier that uses a Naïve Bayesian model for fast elimination, and a pseudo-3D CNN classifier for a more precise foreground detection. Instead of building a model from scratch based on user interaction, the proposed pseudo-3D CNN is trained on a pre-built dataset, which provides a good initialization for interactive fine-tuning and minimizing user bias. The refined model is used to correct the miss-classification that fully explores user interactions at local level. Table 1 lists a number of interactive segmentation techniques that are closely related to the proposed method and provides a comparison in terms of visual feature, delineating technique, optimization method, and user interaction.

2.2 Shape Representation and Regularization

More often than not, interactive segmentation methods utilize some forms of shape representation to generalize the foreground object or region, e.g. active contour, active shape, and active surface models [33, 34, 35, 18, 36]. Broadly, these representations can be categorized as parametric models and geometric models. Parametric models, such as [37, 38, 35, 18, 36], use explicit representations and deform the curves and surfaces explicitly in Lagrangian space, whereas geometric models [39, 40, 34, 19] embed curves and surfaces into the level set function in higher dimension and implicitly evolve them in Eulerian space [41, 42, 43]. Implicit models have more topological flexibility as the shape embedded in a higher dimensional space can split, merge, and vanish more naturally in shape evolution that is driven by a time-dependent Partial Differential Equation (PDE). The data term is typically formulated based on the appearance of the foreground and often heavily rely on sharp edges and homogeneous regions, e.g. [18, 19]. Curve or surface evolution in implicit models is generally solved numerically using finite difference schemes. Periodic reinitialization or reconditioning of the implicit function is necessary to prevent numerical errors contaminating the solution. These fully implicit methods thus are limited to certain topological changes, e.g. unable to develop new contours or surfaces away from exiting ones, and lack the ability to maintain correspondence in shape evolution. Moreover, any shape correspondences, for instance through control points, in their original parametric forms are lost in the process of embedding into fully implicit functions. This also means that it is not possible to directly or interactively manipulate the shape using these implicit models since the control points and their associated regularization are not maintained during

embedding.

There are several attempts to combine implicit representation with explicit representation by approximating the high dimensional embedding function in parametric form, i.e. Parametric Implicit Representation (PIR). Based on the choice of parametrization, PIR can be broadly divided into polynomial and Radial Basis Function (RBF) based approach, both of which can be further categorized into globally and locally supported methods. Table. 2 lists some representative PIR approaches that are used for data reconstruction and image segmentation. The segmentation work using PIR was reported in [25], where a continuous representation of the level set function is parametrized using globally supported Thin-Plate RBF. The deformation is driven by external image force and a constant expansion force, and it moves the locations of zero RBF constraints towards the boundaries of objects. However, it is an incomplete solution, as the location of RBF controls are updated during each iteration while coefficients are fixed, where periodic interpolation is required to reinitialize the implicit function in order to ensure the continuity of the embedding function.

Xie *et al.* [26] overcame the numerical intractability issue by fixing the location of RBF centers. The formulation of coefficient based deformation can then be derived, where level set PDE problem is converted to an Ordinary Differential Equation (ODE) and reinitialization is no longer needed. Later, Paiement *et al.* [27] showed such strategy can be used to solve segmentation and interpolation problems jointly when image data is partially missing. However, computational complexity is a major limitation of globally supported RBF approaches [25, 27, 26]. This is the same for polynomial fitting based PIR [29], as these methods require decomposition of a large and dense kernel matrix that is computationally expensive. Gelas *et al.* [28] introduced compactly supported RBF (Wendland's RBF [44]) to PIR, where a sparse linear system is obtained. The computational complexity of sparse matrix factorization is $O(nzf)$ [45], where nzf is the number of non-zero factors, whereas the complexity for dense ones is $O(N \log N)$. Bernard *et al.* [30] proposed variational B-spline level set that approximates level set function using a number of B-spline basis. It shows that parametric representation can be deformed as a sequential 1-D convolution. Rouhani *et al.* proposed an Implicit B-spline Surface (IBS) based surface reconstruction method for point cloud [31] and showed its feasibility in solving shape registration problem [32]. These methods demonstrate that PIR can be used to approximate level set functions through parametric interpolation and avoid developing numerical errors. However, over-smoothing is one of the main drawbacks of these types of representation, that leads to loss of geometrical details.

It is worth noting that geometrical complexity has not been explicitly considered by previous approaches. In this work, we propose a novel semi-implicit representation, i.e. Non-Uniform Implicit B-spine Surface (NU-IBS), which measures complexity of local topology using scale weighted wavelet coefficients. Denser and more compact bases are used at regions with more intricate structures. Level set based PDE is transformed to an ODE problem, where formulation of coefficient deformation is derived from region based information. The proposed method is equipped with an efficient foreground classifier, where delineation does not rely on hypotheses on boundary, region homogeneity, or reliable initialization.

3 Proposed Method

We propose a unified segmentation framework that utilizes a novel semi-implicit representation which integrates regularization into boundary delineation. The term 'semi-implicit' refers to a shape representation that has explicit, parametrical control points (*i.e.* knots) but over the embedded function that implicitly represents the shape (*i.e.* zero level-set). Segmentation is carried out iteratively using interactive boundary delineation and segmentation regularization. The boundary delineation uses voxel-wise region classification followed by interactive segmentation using adaptive learning. NU-IBS is proposed to allow both flexible shape modeling and shape regularization. The pipeline of our method is illustrated in Fig. 1.

3.1 Foreground-Background Delineation

In this section, a two-stage cascade classifier that consists of a Naïve-Bayesian model and a CNN is introduced to delineate the target objects from volumetric image data.

3.1.1 Intensity-based Naïve-Bayesian Classifier

Given $\mathcal{V}$, a set of $\mathbf{N}$ training voxels, each sample in $\mathcal{V}$ is a double tuple defined as $\mathbf{v}_i \in \mathcal{V}$ and $\mathbf{v}_i := \langle \mathbf{t}_i, \mathbf{c}_i \rangle$, where i is the index for the training sample, $\mathbf{t}_i$ is a scaled integer intensity within the range of $[0, 255]$, and $\mathbf{c}_i \in \{0, 1\}$ is its corresponding binary category label that indicates either background ($\mathbf{c}_i = 0$) or foreground ($\mathbf{c}_i = 1$). Hence, the likelihoods of background and foreground can be empirically estimated using two GMMs with K Gaussian components as follows:

$$\mathcal{P}\ (t|C) = \sum_{k=1}^{K} a_k\ \mathcal{N}(t, \mu_k, \sigma_k^2|C), \quad C \in \{0, 1\} \tag{1}$$

where the mixture weights a, means μ, and stand deviations σ of K Gaussian components can be obtained via Expectation Maximization (EM) given the observations from training dataset. In our case, the foreground and background likelihoods are equivalent to two probability density functions of GMMs, where the parameters of their Gaussian components are estimated

independently on two sets of training samples from the distinct categories. Hence, the Naïve Bayesian classifier can be constructed via choosing an appropriate prior probability $\mathcal{P}(C)$ for each category, and then applying Bayesian rule. For the stage classifier in a cascade framework, it is sensible to scarify the fallout rate to some extent in order to obtain a high recall rate, which can be achieved by setting a biased prior probability towards foreground category. Such strategy compensates the limitation of lacking positive evidence for general detection problem, especially for those extremely unbalanced datasets. It levels the biased data distribution to some extent. Hence, the classifier can be constructed via computing the posterior probabilities of t for all $C \in \{0, 1\}$ given the prior probabilities, and then mapping t to the category label which maximizes the posterior, where t is the intensity value of a target voxel.

3.1.2 Pseudo-3D CNN Classifier

Table 3: The parameter settings of the key components of proposed Pseudo-3D CNN network.

BLK	**Type**	**Parameter**
(a)	C1, C2	13×13 patches from coronal view at two scales
	S1, S2	13×13 patches from sagittal view at two scales
	A1, A2	13×13 patches from axial view at two scales
(b)	Conv. 3	16 (3×3) Conv. filters with stride of 2 pixels
	BNorm	Batch Normalization
	ReLU	Rectified Linear Unit activation function
Output size: 7×7×96		
(c).B5	Conv. 5	192 (5×5) Conv. filters with stride of 2 pixels
	BNorm	Batch Normalization
	ReLU	Rectified Linear Unit activation function
(c).B3	Conv. 3	192 (3×3) Conv. filters with stride of 1 pixel
	BNorm	Batch Normalization
	ReLU	Rectified Linear Unit activation function
	Conv. 3	192 (3×3) Conv. filters with stride of 2 pixels
	BNorm	Batch Normalization
	ReLU	Rectified Linear Unit activation function
Output size: 4×4×384		
(d)	Ave. Pool	4×4 average pooling filter
	FC. 2	Fully Connected layer with 2 outputs
	Softmax	Softmax layer for binary classification

Intensity alone is not sufficient to distinguish the foreground object from background. Inspired by the commonly used Multi-Planar Reconstruction (MPR) in volume rendering, a Pseudo-3D CNN classifier is proposed using the local image patch from three perpendicular panels at multi-scales to precisely identify the foreground object from the hypotheses given by the previous stage. Two-dimensional image patches are a set of appearance projections of the original 3D geometry data from different view angles at a certain location that is within the volume. In most cases, the information from a single projection is ambiguous and biased. However, the uncertainty can be significantly reduced when more projections are available and integrated, especially from uncorrelated views. The proposed pseudo-3D CNN learns primitive features from three orthogonal views independently, which are then aggregated to further generalize abstract descriptors to represent foreground and background elements where a compact classification boundary can be found. Different to full 3D CNN, e.g. [46], which computes 3D convolutional features from a localized volumetric data, the proposed pseudo-3D CNN applies 2D convolutional operators to the images sampled from coronal, sagittal and axial views that centered at the target voxels.

The proposed method has much less processing computational complexity ($O(n^3)$ for full 3D *vs.* $O(3n^2)$ for Pseudo-3D), and it is thus more efficient in terms of training and testing.

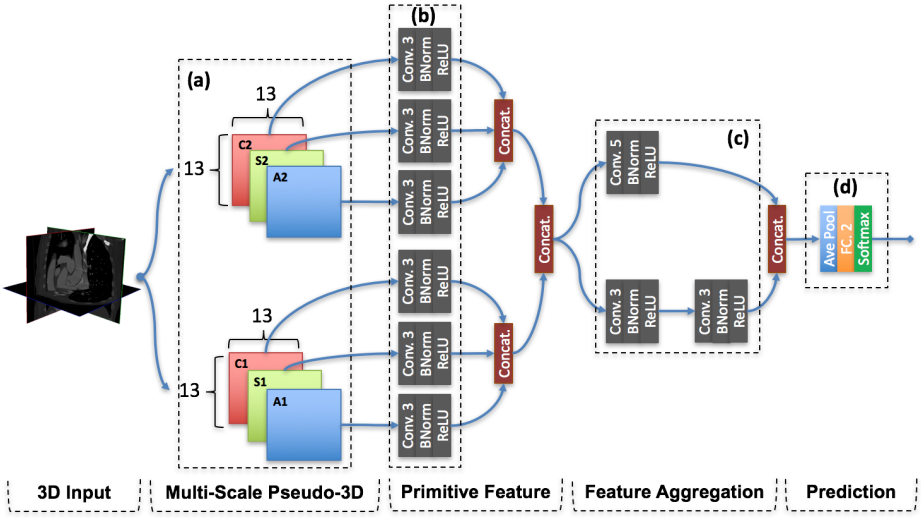


Figure 2: The network architecture of Pseudo-3D CNN classifier which consists of four components as follows: (a) multi-scale Pseudo-3D sampling, (b) primitive feature extraction, (c) feature aggregation and generalization, and (d) foreground-background prediction.

The proposed network consists of four components: (a) multi-scale pseudo-3D sampling, (b) primitive feature extraction, (c) feature aggregation and generalization, and (d) foreground-background prediction. Given a voxel location, block (a) constructs the image patches with size of $W \times H$ pixels from coronal, sagittal and axial views using MPR sampling at multiple scales $\{S_1 ... S_n\}$. Therefore, there are in total $3 \times |S|$ images fed into the networks as inputs (in our case, $W = H = 13$, and $|S| = 2$). In block (b), each primitive feature extractor includes a 3×3 convolutional filtering layer, a batch normalization layer, and a Rectified Linear Unit (ReLU) layer connected consecutively. The primitive feature extraction process is applied to individual images per view per scale, where the learned responses of kernel filters are joined together via channel-wise concatenation. The details of network architecture are illustrated in Fig. 2, and the parameter settings of the key components are listed in Table 3.

Block (c) consists of two branches, a single feature extractor with a 5×5 convolutional layer and two consecutive feature extractors with two 3×3 convolutional layers. Two branches have the same size of receptive field (5×5 pixels) and combine the features channel-wise at the end, whereas the depths of feature abstraction are different (one level for the top branch and two levels for the bottom branch, see Fig. 2 and Table 3). This strategy enforces feature aggregation from different abstraction levels [47]. No pooling layer is used for feature extraction blocks (b & c), spatial down-sampling is applied by setting the stride for convolutional filter to two with boundary padding. Note that for the bottom branch of block (c), only the stride of the last feature extractor is set to two pixels in order to achieve the consistent spatial resolution with the top branch. Therefore, the feature outputs of block (b & c) have the spatial resolutions of 7×7 pixels and 4×4 pixels, respectively. In block (d), an average pooling layer with the filter size of 4×4 pixels is used to reduce the spatial resolution to one pixel, whilst the features are encoded within the channels. Then, a fully connected layer with two output nodes and one *Softmax* prediction layer are followed to perform binary classification. For a given voxel location, the pseudo-3D CNN network encodes the aggregated features using the perpendicular perspective projections from multi-scales, where a set of rich descriptors of local appearances and geometrical structures are hierarchically extracted and then used in discriminative analysis for segmentation.

3.1.3 Localized Interactive Segmentation

For intuitive and minimal user input, we use guiding strokes as means for interactive segmentation. For simplicity, the voxels labeled by foreground strokes are considered as positives and those labeled as background are treated as negatives. Since the user interaction is primitive or minimal, it is crucial that those user strokes as additional ground truth data are effectively translated into useful supervision knowledge on-the-fly to improve segmentation.

The proposed pseudo-3D CNN classifier is trained using Backward Propagation of Errors (Back-Prop) with Stochastic Gradient Descent (SGD) and thus it can be adapted on-the-fly whenever new training data sampled from the guiding strokes is available. However, adaptive learning with limited training samples is often sensitive to newly added training samples in adjusting feature mining, feature selection, and decision boundary [48]. Training under such circumstance with SGD will likely result in poorer performance over time. We introduce three training strategies in order to tackle this issue of oscillation in learning and trade-off between hard outliers and common population. Firstly, the guiding strokes provided by

user are largely from the regions that are previously miss-classified. Fine-tuning the classifier only with patches extracted from the strokes leads to large bias towards a relatively small number of hard samples. Therefore, in addition to the training voxels labeled by the user, a number of pseudo-3D patches are sampled from the initial training data for fine-tuning the classifier. Model refinement will not be dominated by the training data given by the guiding strokes, and the gradient of each mini-batch is corrected to some extent towards the direction which also favors all patch variations. Secondly, in order to prevent breaking the well trained decision boundary for the majorities, a relatively lower learning rate, and a smaller number of training epochs are used to fine-tune the model, which ensures a stable gradient descent optimization, and avoids adapting the model to over-fit those outliers [49]. Thirdly, we observe that the patterns of hard outliers are strongly correlated with their physical locations. In other words, miss-classifications with similar visual and geometrical features are distributed locally. Accordingly, we propose so-called localized refinement strategy that only re-evaluate the local regions on stroke trajectories. In this paper, re-classification with fine-tuned classifier is merely applied to the sub-volume that are $k \times k \times k$ neighbors of the voxels along the guiding strokes. This procedure is performed iteratively until satisfying result is achieved. Algorithm 1 shows the detailed steps of proposed localized interactive segmentation.

Algorithm 1: Localized Interactive Refining

Input : $\mathcal{C}$ is a trained pseudo-3D CNN classifier.
Input : $\mathcal{V}$ is a 3D volumetric image.
Input : $\mathcal{D}_s$ is the binary classification of $\mathcal{V}$ given $\mathcal{C}$.
Output: $\mathcal{D}_i$, $\mathcal{S}_i$ are the refined binary classification and the confidence score of $\mathcal{V}$ respectively.

$\mathcal{D}_i \leftarrow \mathcal{D}_s$, Initiate the classification result;
$n \leftarrow 0$, Initialize the iteration counter;
while *there is user provided foreground strokes* $\mathcal{F}$ *and background strokes* $\mathcal{B}$ **do**
  $n \leftarrow n + 1$, increase the iteration counter;
  $\mathcal{P}_o \leftarrow$ randomly sample foreground and background patches from the original dataset;
  $\mathcal{P}_i \leftarrow$ sample foreground and background patches and labels along user strokes $\mathcal{F}$ and $\mathcal{B}$;
  $\mathcal{C}_i^n \leftarrow$ fine-tune pre-trained CNN classifier $\mathcal{C}$ using $\mathcal{P}_o$ and $\mathcal{P}_i$ with a lower learning rate $\mathcal{L}_i$;
  $\mathcal{V}_i^n \leftarrow$ find irregular sub-volume which contains the voxels that are the k-neighbor of the guiding strokes $\mathcal{F}$ and $\mathcal{B}$;
  $\mathcal{D}_i^n, \mathcal{S}_i^n \leftarrow$ classify and score the sub-volume $\mathcal{V}_i^n$ using fine-tuned model $\mathcal{C}_i^n$;
  $\mathcal{D}_i, \mathcal{S}_i \leftarrow$ merge the refined classification results $\mathcal{D}_i^n$ and confidence scores $\mathcal{S}_i^n$;
end
return $\mathcal{D}_i$ *and* $\mathcal{S}_i$;

3.2 Shape Representation

Given a binary volume, the proposed NU-IBS shape representation uses a set of parametric basis functions that have non-uniform local supports to approximate or interpolate an implicit function for the shape. It is first embedded in an implicit representation based on a signed distance function, followed by approximation using non-uniform B-spline patches in a parametric form. A generic B-spline implicit representation can be formulated as

$$\mathcal{L}(\mathbf{X}) = \mathcal{C}^T \; D(\mathbf{X}) \tag{2}$$

where $\mathbf{X} \in \mathbf{R}^3$ denotes the control knots in a three-dimensional space (represented by xyz-coordinates), $D(.)$ represents the B-spline basis vector given $\mathbf{X}$, $\mathcal{C}$ is the coefficient vector for all B-spline bases, and $\mathcal{L}$ is the approximated signed distance function or level-set function. Next, we present the formulation for uniform implicit B-spline surface, followed by non-uniform expansion using density mapping of control knots that is based on the estimation of shape complexity. We show that constructing the NU-IBS representation is equivalent to solving a non-linear least square problem and the surface can then be reconstructed via interpolation given its implicit parametric formulation.

3.2.1 Uniform Implicit B-spline Surface

For uniform implicit representation with cubic splines, we use four third-degree polynomial blending functions as follows:

$$\begin{aligned} b_0(u) &= (1-u)^3/6 \\ b_1(u) &= (3u^3 - 6u^2)/6 \\ b_2(u) &= (-3u^3 + 3u^2 + 3u + 1)/6 \\ b_3(u) &= u^3/6 \end{aligned} \tag{3}$$

where u is normalized spatial offset from a given knot and b is the value of blending function. In the case of 3D, the knots are placed uniformly along three different axes. We use u_i, v_j, w_k to denote the normalized spatial offset from the i-th, j-th,

k-th knots along xyz axes, respectively. Let r, s, t be the indices of blending functions defined in (3) for each axis, N be the number of knots that are uniformly placed, and $c_{i,j,k}$ be the coefficient for knot (i, j, k). Given a local control point $Knot_{ijk}$ in 3D, the indices of knot and corresponding u_i, v_j, w_k for each axis can be computed as follows:

$$\begin{aligned} \delta &= 1/(N-3) \\ i = \lceil x/\delta \rceil, \quad u_i &= x/\delta - \lfloor x/\delta \rfloor \\ j = \lceil y/\delta \rceil, \quad v_j &= y/\delta - \lfloor y/\delta \rfloor \\ k = \lceil z/\delta \rceil, \quad w_k &= z/\delta - \lfloor z/\delta \rfloor \end{aligned} \tag{4}$$

where $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ denote ceiling and floor rounding operators, respectively. Therefore, the level-set function $\mathcal{L}$ can be approximated based on (3) and (4) as:

$$\mathcal{L}(\mathbf{X}) = \sum_{r,s,t=0}^{3} c_{i+r,j+s,k+t}\ b_r(u_i) b_s(v_j) b_t(w_k) \tag{5}$$

Fig. 3 (a.1) shows a 1D example of cubic B-Spline basis functions that are made out of scaling and translating uniform blending functions, and Fig. 3 (a.2) shows the unweighed uniform kernel functions that are computed using (5) with constant coefficients ($\forall c = 1$).

3.2.2 Non-Uniform Expansion

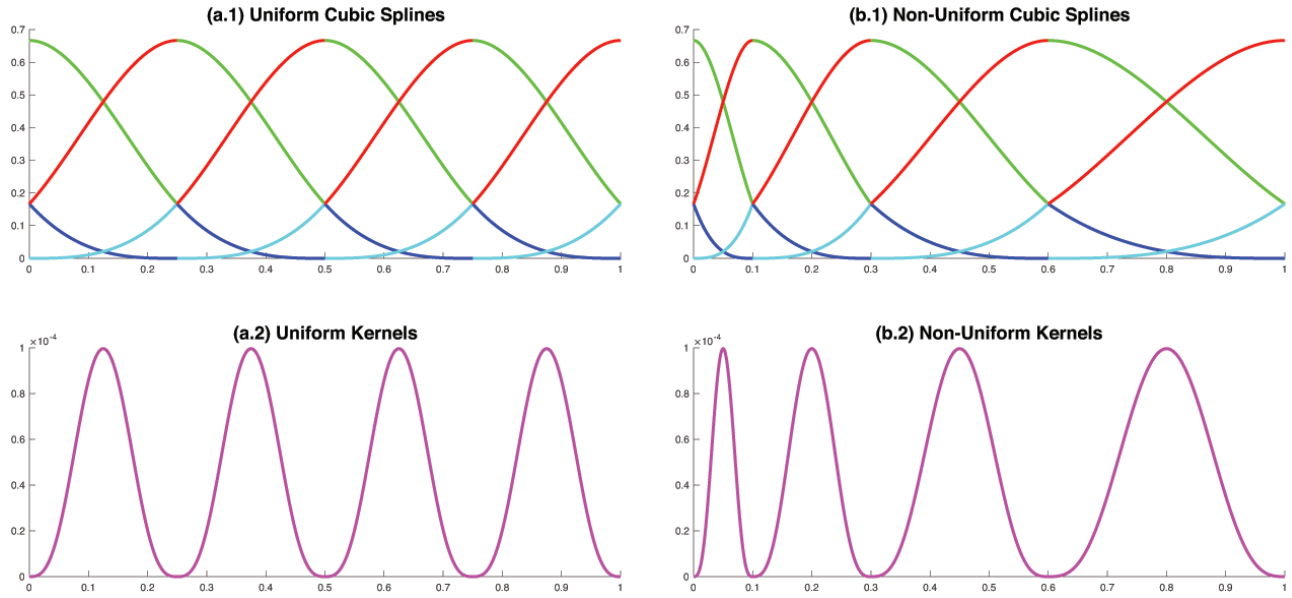


Figure 3: (a.1) A 1D example of cubic B-splines basis functions that are made out of scaling and translating the uniform blending functions. (a.2) An example of unweighted uniform kernel functions. (b.1) A 1D example of cubic B-Spline basis functions that are made out of scaling and translating the non-uniform blending functions. (b.2) An example of unweighted non-uniform kernel functions.

Uniform model places the control points evenly in the defined intervals. However, for shapes with varying degree of complexity, non-uniform distribution of control points is more desirable, with more densely populated control points positioned in regions of complex structure, e.g. high curvature, and much more sparser distribution for regions with limited geometrical features. We thus introduce a novel non-uniform expansion to the proposed semi-implicit representation. Fig. 3 (b.1) shows a 1D example of cubic B-Spline basis functions that are made of scaling and translating non-uniform blending functions, and Fig. 3 (b.2) shows the unweighed non-uniform kernel functions that are computed using (5) with constant coefficients ($\forall c = 1$). In Fig. 3 (b.2), the densely distributed control points (on the left) that has small support radius provide more compact representation compared to the sparse ones (on the right).

As before, the shape to be represented is embedded in the zero level set of an implicit function, $\mathcal{L}$ as defined in Equ. 2 and 5. Surface regions with high curvature, large oscillation, and high frequency in the embedded signed distance function are considered to be complex regions, where more control points are required.

Thus, a scale weighted complexity estimation method is proposed to determine the density of control points or knots. Fig. 4 (a) shows a 1D signal that is constructed using a set of sine functions that have different frequencies. Fig. 4 (b) shows

the heat map of continuous wavelet coefficients of the signal in multiple scales. Given an implicit shape representation $\mathcal{L}_0$, the density of shape complexity along one axis can be estimated via marginalizing the scale weighted amplitudes of Gaussian wavelet responses over other two axes, as follows:

$$\begin{aligned}
\mathcal{W}_x &= \sum_{y=1}^{Y} \sum_{z=1}^{Z} \sum_{m=1}^{M} \frac{1}{m} ||\mathcal{L}_0(:, y, z) \otimes \mathcal{K}_{gaus}||_1 \\
\mathcal{W}_y &= \sum_{x=1}^{X} \sum_{z=1}^{Z} \sum_{m=1}^{M} \frac{1}{m} ||\mathcal{L}_0(x, :, z) \otimes \mathcal{K}_{gaus}||_1 \\
\mathcal{W}_z &= \sum_{x=1}^{X} \sum_{y=1}^{Y} \sum_{m=1}^{M} \frac{1}{m} ||\mathcal{L}_0(x, y, :) \otimes \mathcal{K}_{gaus}||_1
\end{aligned} \tag{6}$$

where M is the number of scales, $\mathcal{K}_{gaus}$ is the kernel filter of Gaussian wavelet, and $\otimes$ is the convolution operator. The amplitudes ($L1$ norm) of wavelet coefficients are weighted by the reciprocal of the scales, which lowers the contributions of detected signal oscillation in large scales, while capturing details in small scales. Therefore, the distribution of shape complexity can be interpreted as the density histogram of subtle changes of geometrical structures along certain axis over the whole volume. We assume N B-splines that divide the definition domain into $N-3$ intervals where the knots are placed at the intersections. NU-IBS divides the intervals according to density histogram $\mathcal{W}_x, \mathcal{W}_y$, and $\mathcal{W}_z$, which ensures that each interval has even accumulated density, where such mapping can be conveniently obtained by using histogram equalization. For NU-IBS, the uniformly distributed knots, as computed in (4), are replaced by this adaptive mapping method, and the basis vector D can be constructed accordingly. By doing so, the regions that have complex geometrical structures are approximated using more B-spline patches with compact supports, when the total number of B-splines is fixed.

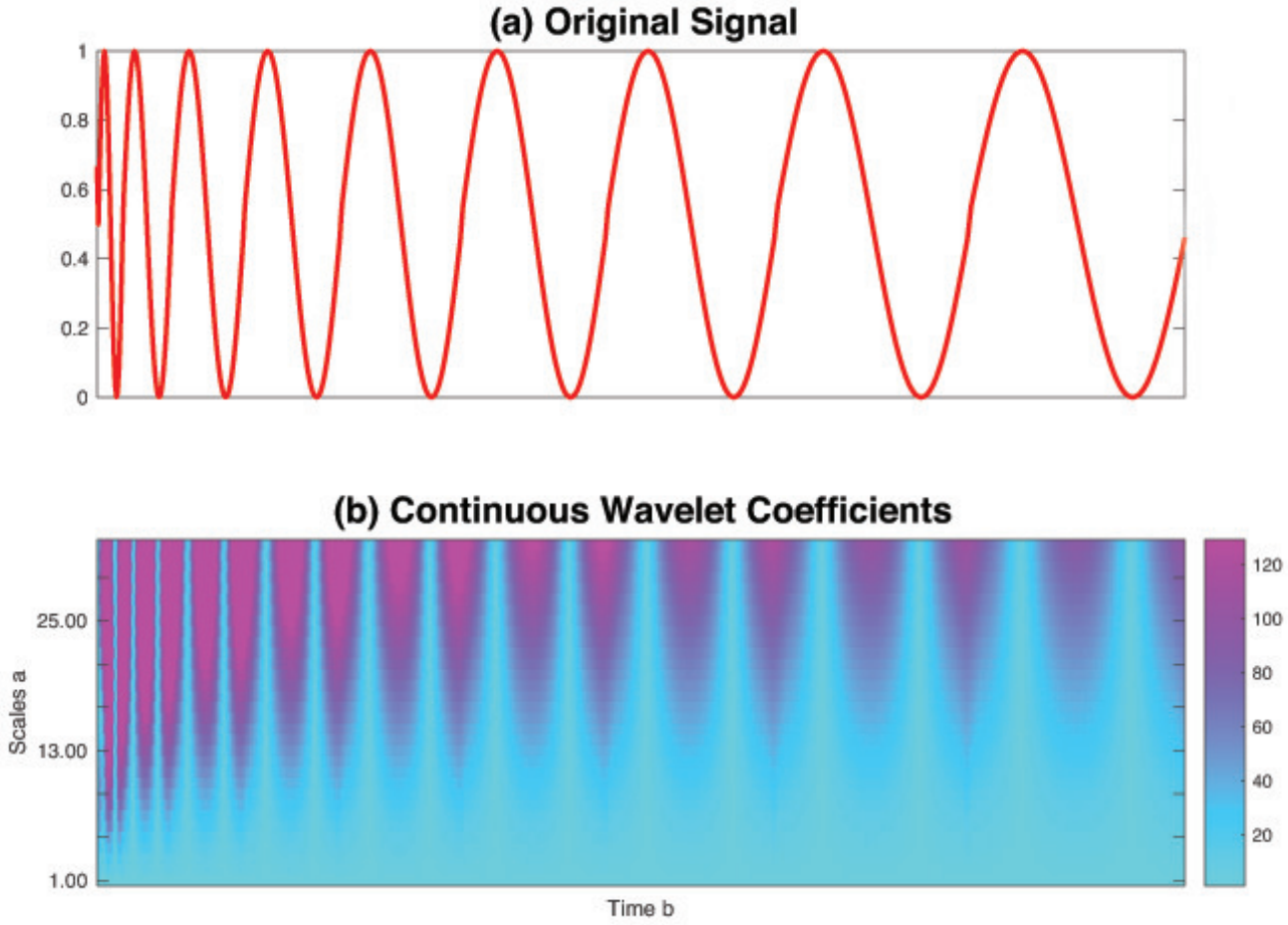


Figure 4: (a) A 1D signal is constructed using a set of sine functions that have different frequencies. (b) The heat map of continuous wavelet coefficients of the signal given in (a).

3.2.3 Surface Parameterization and Construction

The level set function is obtained using signed distance transform of the binary volume given by foreground-background delineation stage. In order to ensure the numerical stability, we follow the convention that distance values inside the object are positive, and the values at outside are negative. The surface parameterization is equivalent to solving the following non-linear least square problem with ridge regularization:

$$\begin{aligned} \mathcal{C}^* &= \arg\min_{\mathcal{C}} \ \{||\mathcal{L}' - \mathcal{L}(\mathbf{X})||_2 + \mu(\mathcal{C}^T \ \mathbf{I} \ \mathcal{C})\} \\ &= \arg\min_{\mathcal{C}} \ \{||\mathcal{L}' - \mathcal{C}^T \ D(\mathbf{X})||_2 + \mu(\mathcal{C}^T \ \mathbf{I} \ \mathcal{C})\} \end{aligned} \tag{7}$$

where the μ is the regularization parameter, and $\mathbf{I}$ is the identity matrix. Other regularization term can also be used such as curvature penalty. Given a uniformly sampled sub-volume $\mathbf{S}$ from $\mathcal{L}'$, the vectorized distance values $\mathbf{B}_s$ and basis matrix $D_s(\mathbf{X})$ can be constructed by concatenating the basis vector of corresponding points in $\mathbf{S}$ row-by-row, such that the approximated solution can be found as follows:

$$\begin{aligned} \mathcal{C} &= D_s^{\dagger}(\mathbf{X}) \ \mathbf{B}_s = (D_s(\mathbf{X})^T D_s(\mathbf{X}))^{-1} D_s(\mathbf{X})^T \mathbf{B}_s \\ \mathcal{C}^* &= (D_s(\mathbf{X})^T D_s(\mathbf{X}) + \mu \ \mathbf{I})^{-1} \ D_s(\mathbf{X})^T \ \mathbf{B}_s \end{aligned} \tag{8}$$

where $D_s^{\dagger}(\mathbf{X})$ denotes the pseudo inverse of $D_s(\mathbf{X})$. The basis matrix $(D_s(\mathbf{X})^T D_s(\mathbf{X}))^{-1}$ is a highly sparse matrix, where much faster factorization algorithms are available compared to dense matrix [50]. Given the parametric representation of a shape, the level-set function can be computed via interpolating the distance field of the shape within the definition domain using (2). Hence, the surface is reconstructed on the zero level-set manifold.

3.3 Segmentation as Region-based Deformation

A binary volume can be obtained using the proposed 2-stage cascade classifier and interactive adaptive segmentation, where each voxel is assigned with either foreground or background label independently. It can be considered as an initial representation for the object of interest. A more precise and compact representation that captures the geometrical and topological features is required. The piecewise constant model has been widely used as a regularization scheme in a variety of tasks, such as denoising [51], image restoration [52], and segmentation [53, 54, 55] . It assumes that the appearance or visual feature of the foreground object is locally homogeneous. A regularization term that imposes this piecewise homogeneity constraint is typically added to the objective function that aims to separate the foreground and background. In our case, the NU-IBS is approximated parametric form of the level set function given a shape, which can be constructed directly from the binary decision volume. In order to impose the geometrical smoothness constrain to the loosely detected object through data support, the classification confidence scores of each voxel are used to construct a PDE derived from the classical Chan-Vese energy functional [56]. Therefore, the shape is deformed via propagating the zero interface Γ, which leads to a solution that partitions the image into two regions with local homogeneity and delimiting the boundaries of foreground object:

$$\begin{aligned} J(\mathcal{L}) = \lambda_1 &\int_{\Omega} \delta(\mathcal{L}) ||\nabla \mathcal{L}|| \ d\mathcal{C} \\ &+ \lambda_2 \int_{\Omega} (\mathcal{S}(\mathcal{C}) - C_1(\mathcal{L}))^2 \cdot u(\mathcal{L}) \ d\mathcal{C} \\ &+ \lambda_3 \int_{\Omega} (\mathcal{S}(\mathcal{C}) - C_2(\mathcal{L}))^2 (1 - u(\mathcal{L})) \ d\mathcal{C} \end{aligned} \tag{9}$$

where u and δ are respectively the Heaviside and Dirac univariate functions, $\lambda_1, \lambda_2, \lambda_3$ are positive hyper-parameters that control the contributions from the surface smoothness, inside and outside homogeneity of partitions, and $\mathcal{L}$ is the approximated level set using B-spline interpolation. $C_1(\mathcal{L}), C_2(\mathcal{L})$ are computed during the interface propagation at each iteration using the following expression:

$$\begin{aligned} C_1(\mathcal{L}) &= \frac{\int_{\Omega} \mathcal{S}(\mathcal{C}) \cdot u(\mathcal{L}(\mathcal{C}, t)) \ d\mathcal{C}}{\int_{\Omega} u(\mathcal{L}(\mathcal{C}, t)) \ d\mathcal{C}} \\ C_2(\mathcal{L}) &= \frac{\int_{\Omega} \mathcal{S}(\mathcal{C}) \cdot (1 - u(\mathcal{L}(\mathcal{C}, t))) \ d\mathcal{C}}{\int_{\Omega} (1 - u(\mathcal{L}(\mathcal{C}, t))) \ d\mathcal{C}} \end{aligned} \tag{10}$$

The general minimization solution of $J(\mathcal{L})$ can be found using variational calculus and gradient descent method [42, 57, 56], as follows:

$$\begin{aligned} \frac{\partial \mathcal{L}(\mathcal{C}, t)}{\partial t} + \mathbf{V}(\mathcal{C}, t) \cdot \delta_{\epsilon}(\mathcal{L}(\mathcal{C}, t)) = 0 \\ \delta_{\epsilon}(x) = \frac{1}{\pi \epsilon \cdot (1 + (\frac{x}{\epsilon})^2)} \end{aligned} \tag{11}$$

where δ_ϵ is a regularized Dirac function. Note that in (9), λ_1 controls the contribution weight of surface smoothness which is already assured by the intrinsic property NU-IBS. We can simply set $\lambda_1 = 0$ and $\lambda_2 = \lambda_3 = 1$. Then, the velocity term is given as:

$$\mathbf{V}(\mathcal{C}, t) = -(\mathcal{S}(\mathcal{C}) - C_1(\mathcal{L}))^2 + (\mathcal{S}(\mathcal{C}) - C_2(\mathcal{L}))^2 \tag{12}$$

By combining (2 7 and 11), the PDE can then be transformed to an ODE with respect to the B-spline coefficients $\mathcal{C}$ of NU-IBS, where the optimal segmentation can be found by iteratively updating $\mathcal{C}$ according to the detection confidence score $\mathcal{S}$ until the steady state is reached. The gradient descent solution is given as:

$$\begin{aligned} \frac{d\mathcal{C}}{dt} &= -\{(D_s(\mathbf{X})^T D_s(\mathbf{X}) + \mu\mathbf{I})^{-1}\ D_s(\mathbf{X})^T \\ &\quad \times(\mathbf{V}(\mathcal{C}, t) \cdot \delta_\epsilon(\mathcal{L}(\mathcal{C}, t)))\} \\ \mathcal{C}_{(n+1)} &= \mathcal{C}_{(n)} + \tau \frac{d\,\mathcal{C}_{(n)}}{dt} \end{aligned} \tag{13}$$

where τ is the step size. A small τ value enables a steady numerical solution and more updating iteration is required to converge.

4 Experimental Results and Analysis

4.1 NU-IBS Representation Evaluation

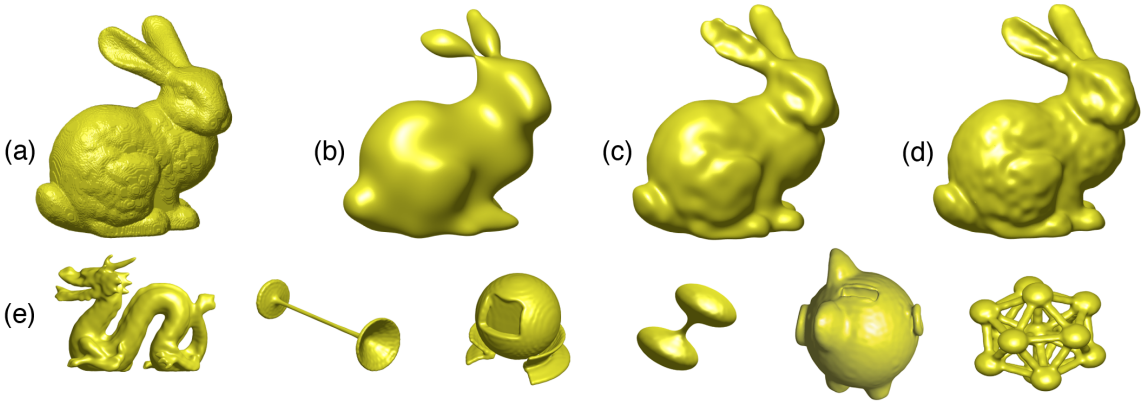


Figure 5: Examples of geometries that are represented using proposed NU-IBS. (a) "bunny" constructed directly from original mesh model. (b) Bunny with 10 B-splines. (c) Bunny with 20 B-splines. (d) Bunny with 30 B-splines. (e) Some other typical examples including "dragon", "floor lamp", "mitsuba", "navalstring", "piggy" and "molecule" (from left to right).

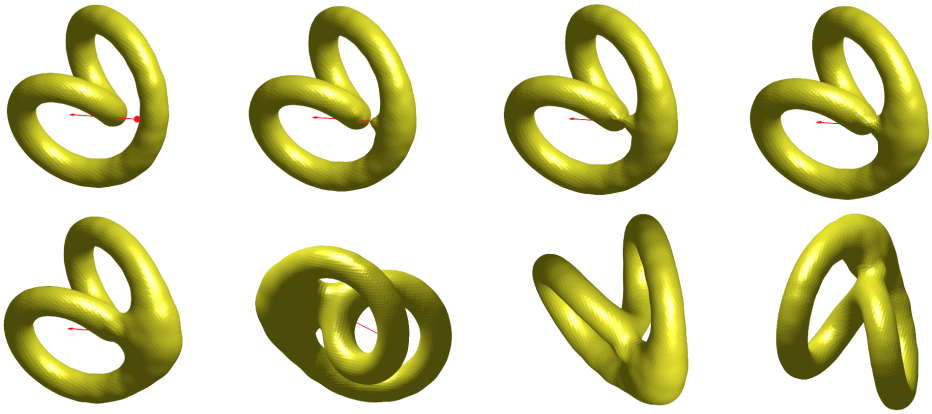


Figure 6: Examples of geometrical manipulation of NU-IBS. The first row shows progressive topological merging by dragging the surface along targeted direction, the second row shows merged surfaces from different perspectives.

Table 4: Quantitative Evaluation of Shape Representation Methods

Support	Linear System	Method	Dice Coefficient	Jaccard Score	Hausdorff Distance
Global	Dense $O(NlogN)$	Thin-Plate [25]	0.9456 ± 0.0379	0.8991 ± 0.0655	6.7478 ± 3.9113
		Multi-Quad [26]	0.9405 ± 0.0448	0.8909 ± 0.0756	6.8251 ± 3.8545
		Gauss [27]	**0.9472 ± 0.0309**	**0.9013 ± 0.0550**	6.8658 ± 4.0749
		Polynomial [29]	0.9265 ± 0.0715	0.8706 ± 0.1149	**6.2344 ± 3.5062**
Local	Sparse $O(N_{non-zeros})$	Wendland [28]	0.8983 ± 0.0635	0.8214 ± 0.1033	9.0405 ± 6.5569
		Uniform BSpline [30, 31, 32]	0.9289 ± 0.0639	0.8734 ± 0.1038	6.7949 ± 4.7125
		Proposed Method	**0.9407 ± 0.0630**	**0.8941 ± 0.1017**	**5.3165 ± 2.9721**

The proposed NU-IBS representation was evaluated qualitatively using a number of classic 3D scans (*i.e.* Stanford Bunny [58] and Dragon [59]), synthetic images (*i.e.* Naval-String and Molecule) and other commonly used 3D models in computer graphics. The binary volume was firstly built by rasterizing the original triangular mesh, within which the breaking parts and holes were filled up and repaired manually. The signed distance field can then be computed directly given the binary volume, and the data points were sampled coarsely from a uniform 3D grid, whereas the control knots were placed irregularly using the proposed shape complexity estimation method described in Sec 3.2.2. Therefore, the NU-IBS coefficients were estimated given sparse data samples and control knots, and the shape was reconstructed by computing the signed distance field of a densely sampled 3D grid.

In Fig. 5, we demonstrate the capability of the proposed shape representation by reconstructing given geometries with finite control points. In the first row, we show that the proposed method interpolates and reconstructs a complex shape with different level of details by varying overall number of control points. It can be observed that more subtle details are captured as the number of control points increases, whilst the computational complexity increases polynomially. The second row shows several examples with large variations in geometry, including thin tubular structures, concavities, and fine connecting structures. For instance, "floor lamp", "navalstring" and "molecule" examples demonstrate that connectivity is well maintained in the embedding and interpolation processes even at a coarse scale due to the use of non-uniform distribution of control knots. Cavity structure is often a problematic case for implicit representations. With explicit parametric control, our method is able to naturally handle such geometries (see "mitsuba", "piggy" and "molecule" in Fig. 5).

Quantitative comparison for all seven different shape representation methods on those 3D geometries, as shown in Fig. 5, is given in Table 4. An equal number control points (30) is used for all experiments. For the proposed NU-IBS, these control points are distributed non-uniformly according to the complexity of local geometry. For all other methods, they are uniformly distributed. Kernel functions with global support generally performs better than locally supported ones, due to additional smoothness. However, local methods are much more efficient in computation and significantly more economical in memory usage, due to their sparse linear systems. Moreover, locally supported representation allows precise, localized editing. Our proposed local methods outperforms all other local techniques and closely matches the best performing global ones. The Hausdorff distance of the proposed method is substantially smaller than all other methods, which strongly suggests the benefit of nonuniform distribution of control knots.

In addition to surface reconstruction, an interactive shape manipulation scheme can also be derived by setting (12) to a constant vector in R^3 space at a localized control knots. Therefore, the surface will deform along the direction of given force and the speed is directly proportional to its magnitude. Fig. 6 shows that shape manipulation can be done by locally deforming the surface along a given direction with a constant force magnitude. The topological flexibility of NU-IBS for shape editing can also be observed (see the top row in Fig. 6), where two parts merge naturally.

4.2 Segmentation Evaluation on 3D CTA Dataset

The proposed segmentation method was quantitatively evaluated on a 3D CTA dataset, which contains 36 volumetric Transcatheter Aortic Valve Implantation (TAVI) scans collected in collaboration with cardiologists from a UK NHS Trust. The number of slices of each scan varies from 500 to 800, while the image size of each slice is 512×512 pixels across all scans, and the voxel dimension is $0.48 \times 0.48 \times 0.62mm^3$. The general Cardiac CT imaging acquisition protocol was followed, while the radiation exposures and acquisition timing were adjusted individually across different patients in order to ensure image quality and meet patient requirements. The original 3D scans contain full torso, which were cropped to contain the whole cardiac regions with a fixed size of 256×256 pixels per slice.

The anatomical structure to be segmented is the aortic system, including ascending aorta, descending aorta, and aorta root that contains the three aortic leaflets in closing positions. The aorta is the large blood vessel that carries oxygen-rich blood from the left ventricle of the heart to other parts of the body, where its root attaches to the heart. The aortic root consists of three aortic valve leaflets and the coronary ostia which are the openings for the coronary arteries. In this paper, we are segmenting the ascending, descending aortas and three valve leaflets which form an arch-like structure (see Fig. 10). An example of 3D TAVI image and its 3D surface rendering are shown in Fig. 7, where the aorta root is highlighted with a circle in orange which corresponds to those shown in Fig. 10. To label the ground-truth, the Region of Interest (ROI)

including aorta root and arch are cropped out from each 3D scan. The root including three valve leaflets, and the arch were labeled slice by slice up to the top of the arch using closed contours. Hence, a binary volume can be constructed where insides of the contours were assigned to 1, and outsides were 0 indicating foreground objects and background respectively.

Segmenting such a delicate structure from large volumes is a challenging task. This aortic system has heterogeneous appearance and detailed geometrical features at local level, e.g. its arch is generally a smooth structure and its root is formed together with three valve leaflets with intricate details. Moreover, there are several structures in upper torso have similar appearances and geometrical features. For instance, the pulmonary artery and superior vena cava exhibit close resemblance and can lead to ambiguities for segmentation.

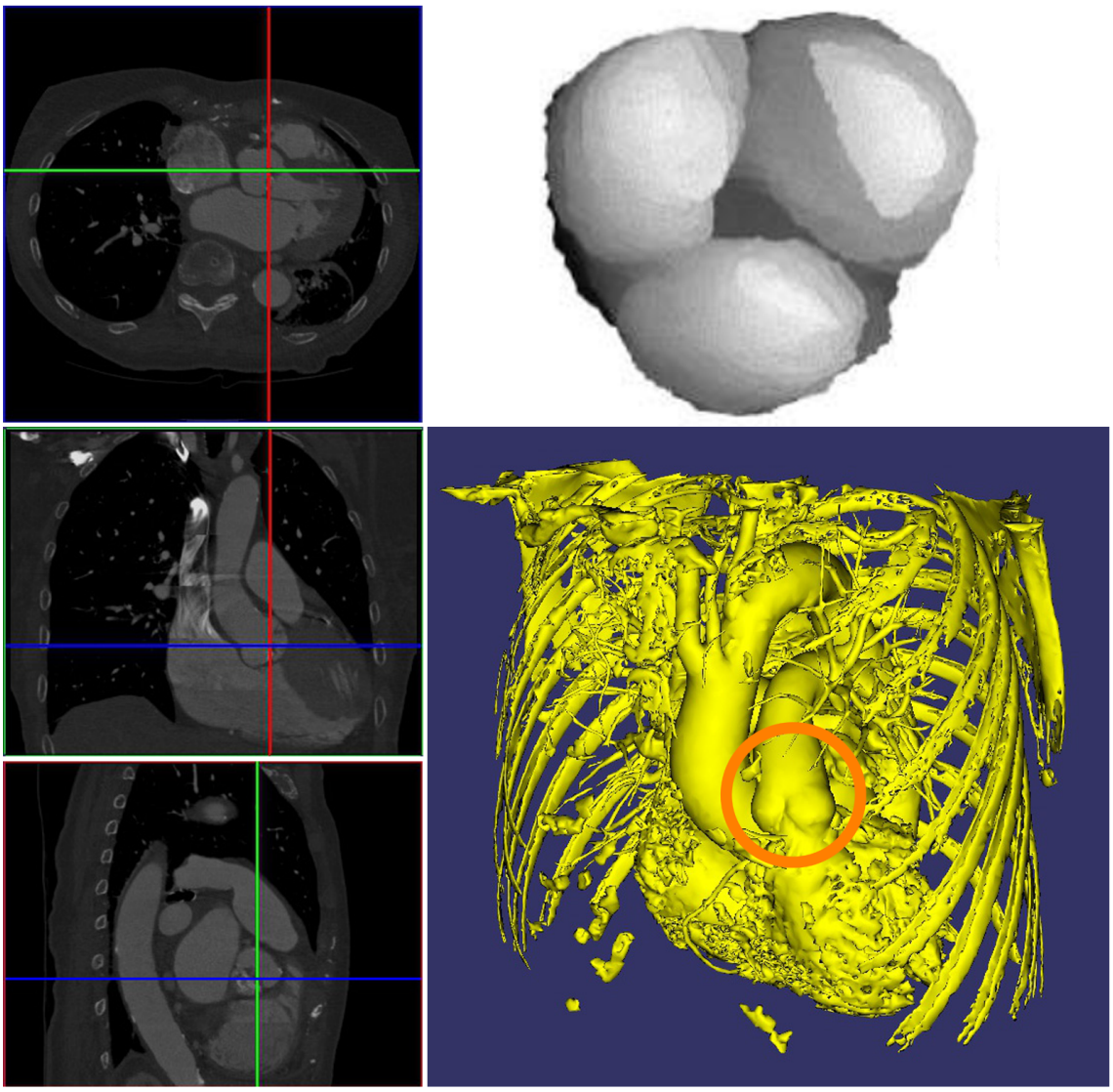


Figure 7: An example of 3D CTA TAVI image from 3 orthogonal views and surface rendering created using *3DimViewer* [60]. The images from the top to the bottom in the left column are axial view, coronal view and sagittal view respectively. The right column shows the mesh model of aorta root (top) and 3D surface rending of the volume (bottom), where the aorta root is highlighted with the organ circle.

We carried out three-fold cross validation for evaluating the segmentation of the aortic system. The intensity value was scaled into the range of $[0, 255]$ given the optimal window size and window level that were provided in the Digital Imaging and Communications in Medicine (DICOM) image meta information. To train the Naïve-Bayesian classifier, 300K foreground and 300K background voxel intensities from each training volume were collected, which was about 6% of the total number of voxels in the whole volume. A GMM with 5 Gaussian components was used, and the conditional posterior probabilities were computed given an even foreground and background prior. Fig. 8 shows three Naïve-Bayesian classifiers that were constructed for different fold tests. The blue and cyan curves are conditional probabilities of foreground and background that are modeled using GMMs, and the red and black curves are posterior probabilities obtained through Bayesian rule given a pre-defined prior. To train a Pseudo-3D CNN classifier, the false positive and the false negative voxels were collected from which the multi-scale Pseudo-3D patches were sampled from each volume. The size of mini batch was 512, the number of

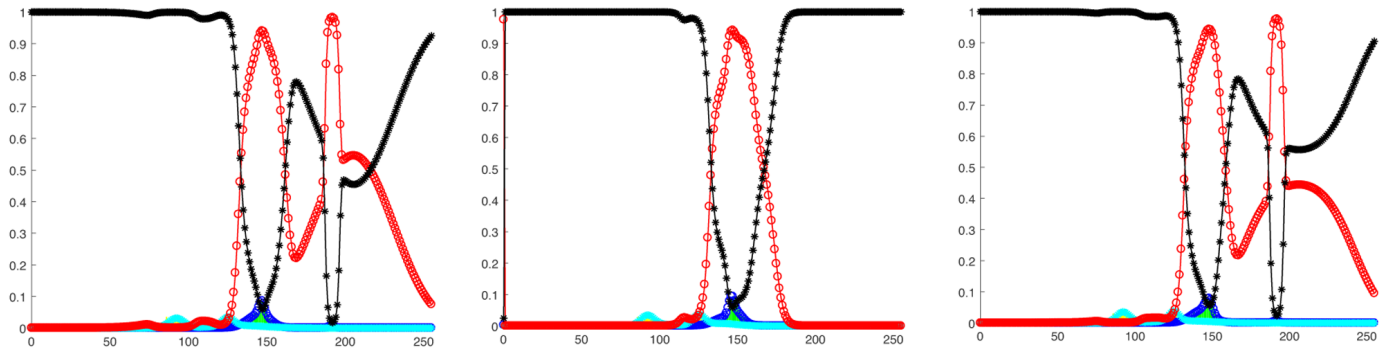


Figure 8: The visualization of three Naïve-Bayesian classifiers trained for different data fold on *3D CTA* dataset. The blue and cyan curves are conditional probabilities of foreground and background that are modeled using GMMs. The red and black curves are posterior probabilities obtained through Bayesian rule given a pre-defined prior.

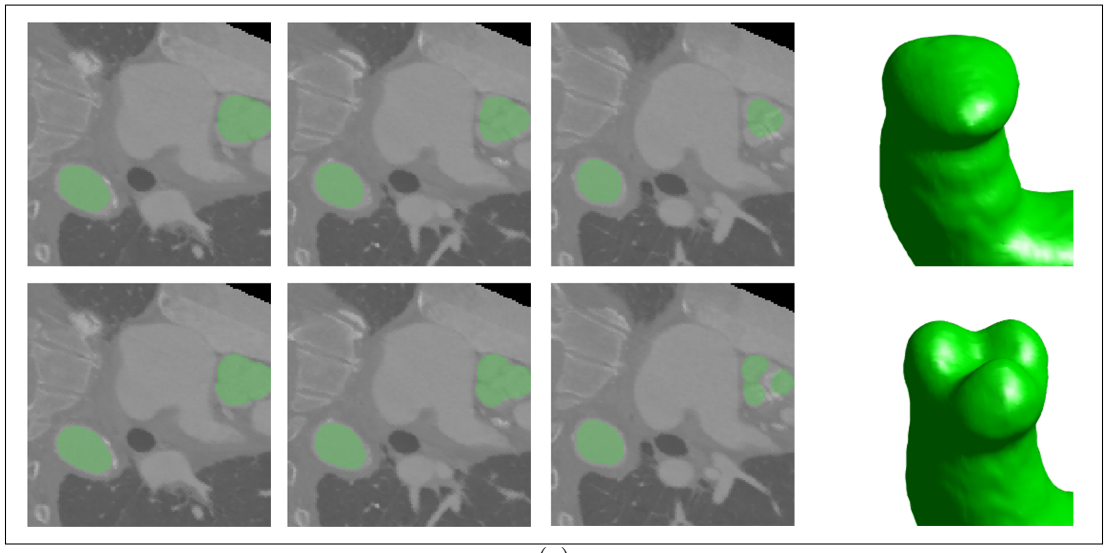
epochs was set to 6, which lead to 63,420 iterations in average. The initial learning rate was set to 0.1, and then divided by a factor of 10 every two epochs. To simulate the localized interactive refining procedure, we randomly selected 1,280 voxels from both false positives and false negatives that were given by the Pseudo-3D classifier as user guiding strokes. In addition, 3,840 voxels were randomly sampled from the original dataset, which was together with the simulated guiding strokes making a fine-tuning dataset with 5,120 samples in total. The learning rate and training epoch for fine-tuning were set to $1e$-5 and 10 respectively to avoid large decision boundary shifting. The localized refining was applied to $9 \times 9 \times 9$ sub-volumes that were centered at the voxels from simulated guiding strokes. The maximum length of the strokes is limited to the diagonal length of the volume.

The classification results of individual stages from different folds are presented in Table 5. The Naïve-Bayesian classifier achieved 93.20% true positive rate in average, with a false positive rate of 9.63%. However, as the majority of the volume is background the false positive number is in fact too high for the segmentation to be useful. Structures that have similar appearances are mistakenly considered as foreground, such as rib cage, pulmonary artery, blood vessels in the lungs, ventricles and atria (see Fig. 10 row (a)). In the next stage, the pseudo-3D CNN classifier dramatically eliminates those false positives by learning the spatial feature in a hierarchical fashion and reduced the false positive rate to a magnitude smaller, i.e. 0.82%. At the same time, the true positive rate reduced by a moderate amount, 8.06%. Generally, the misclassification happens near the outer boundaries of the arch and the tips of leaflets, as shown in blue in Fig. 10 row (b), where those regions either lack sharp edges or are too close to other large blood vessels. The final stage of localized interactive segmentation greatly improved the accuracy. The first iteration alone boosted the true positive rate from 85.14% to 92.16% and reduced the false positive rate even further. Moreover, the localized interactive segmentation prevents contaminating the well classified regions. Fig. 10 rows (c) and (d) show examples of the first round and the last round of iterations, respectively. The false positives (yellow regions) and false negatives (blue regions) are eliminated progressively, especially the example in the second column. There is very minor fluctuation of false positive rate while the segmentation stabilizes, i.e. as the methods approaches the upper limit of the discriminative power of pseudo-3D CNNs . Finally, a connectivity component analysis is applied as a post-processing step to remove any isolated small regions.

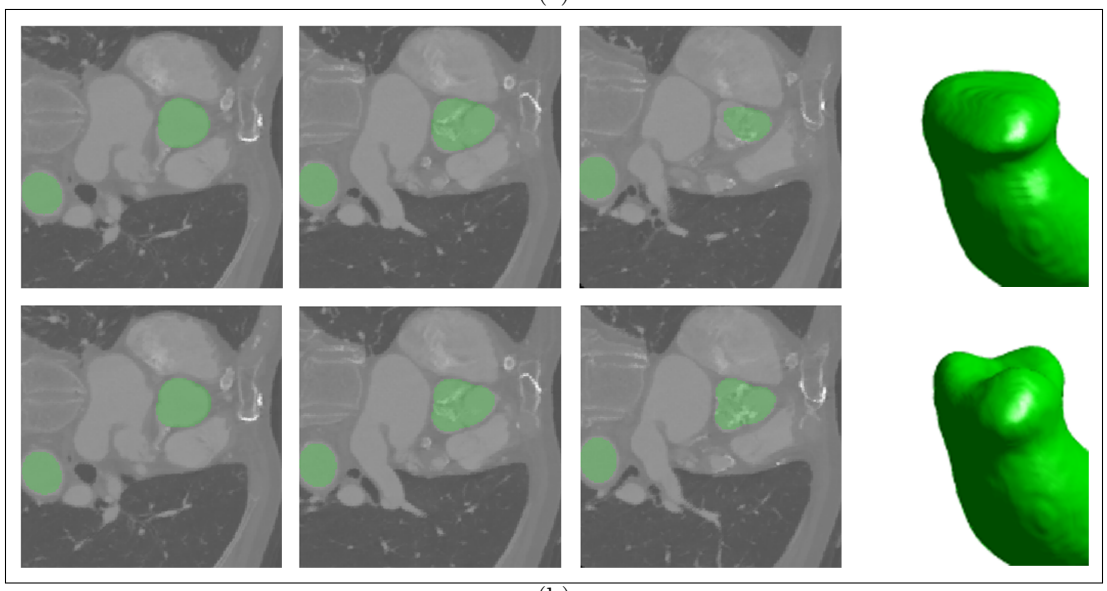
Table 5: Quantitative results on *3D CTA* dataset (TP: true positive, FP: false positive, in %) at each cascade stage and localized interactive segmentation.

	Fold-1		Fold-2		Fold-3		Avg.	
	TP	FP	TP	FP	TP	FP	TP	FP
N-B	94.79	8.65	87.99	9.41	96.83	10.83	93.20	9.63
P-3D	84.96	0.66	81.56	0.86	88.91	0.95	85.14	0.82
Ref-1	93.81	0.44	90.35	0.45	92.31	0.61	92.16	0.50
Ref-2	94.89	0.46	91.14	0.39	93.02	0.64	93.02	0.50
Ref-3	95.30	0.48	92.11	0.40	93.55	0.66	93.65	0.51
Ref-4	95.59	0.49	93.07	0.43	94.02	0.69	94.23	0.54
Ref-5	95.89	0.50	94.06	0.47	94.53	0.71	94.83	0.56

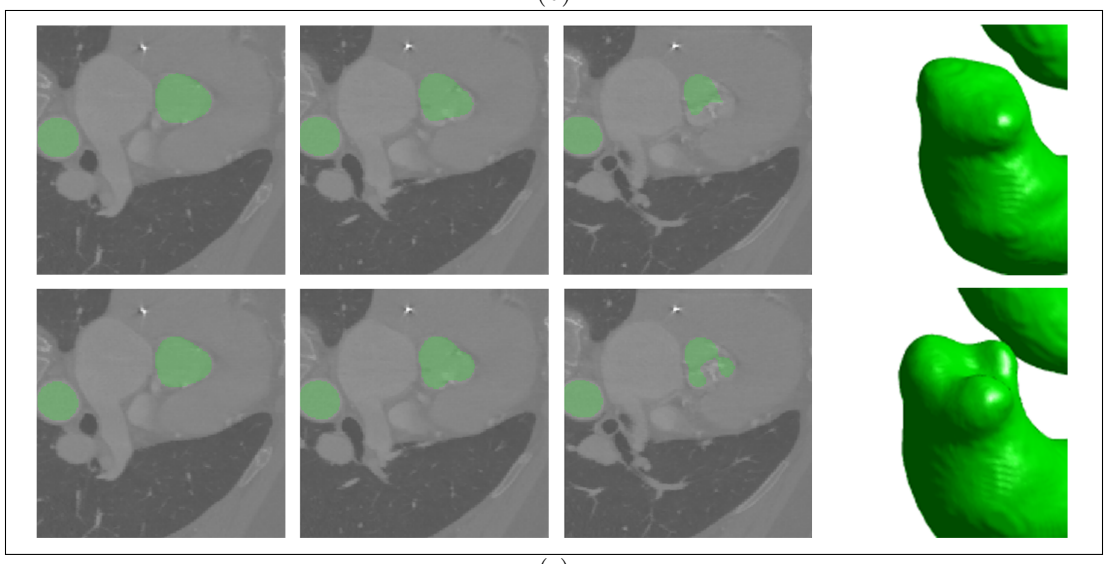
The shape representation of classified aortic vessel was initially constructed using the binary classification volume, and then deformed with regard to the normalized prediction scores until it converges ($\Delta C_1 + \Delta C_2$) <5e-4). The number of maximum iteration was set to 50. The regularization parameters of Heaviside and Dirac functions were set to $1e$-5 and $1e$-1 respectively, and the step size τ was set to $1e$-1 for all iterations. We compared the proposed NU-IBS with uniform IBS using 23 and 28 B-splines with different sampling rates. The quantitative measurements are listed in Table 6 which were calculated



(a)



(b)



(c)

Figure 9: Qualitative comparisons of uniform IBS (the first row of each sub-figure) and proposed non-uniform IBS (the second row of each sub-figure) on *3D CTA* dataset. The uniform method tends to smooth out the geometrical details of aorta valves, which are however preserved very well by our proposed method.

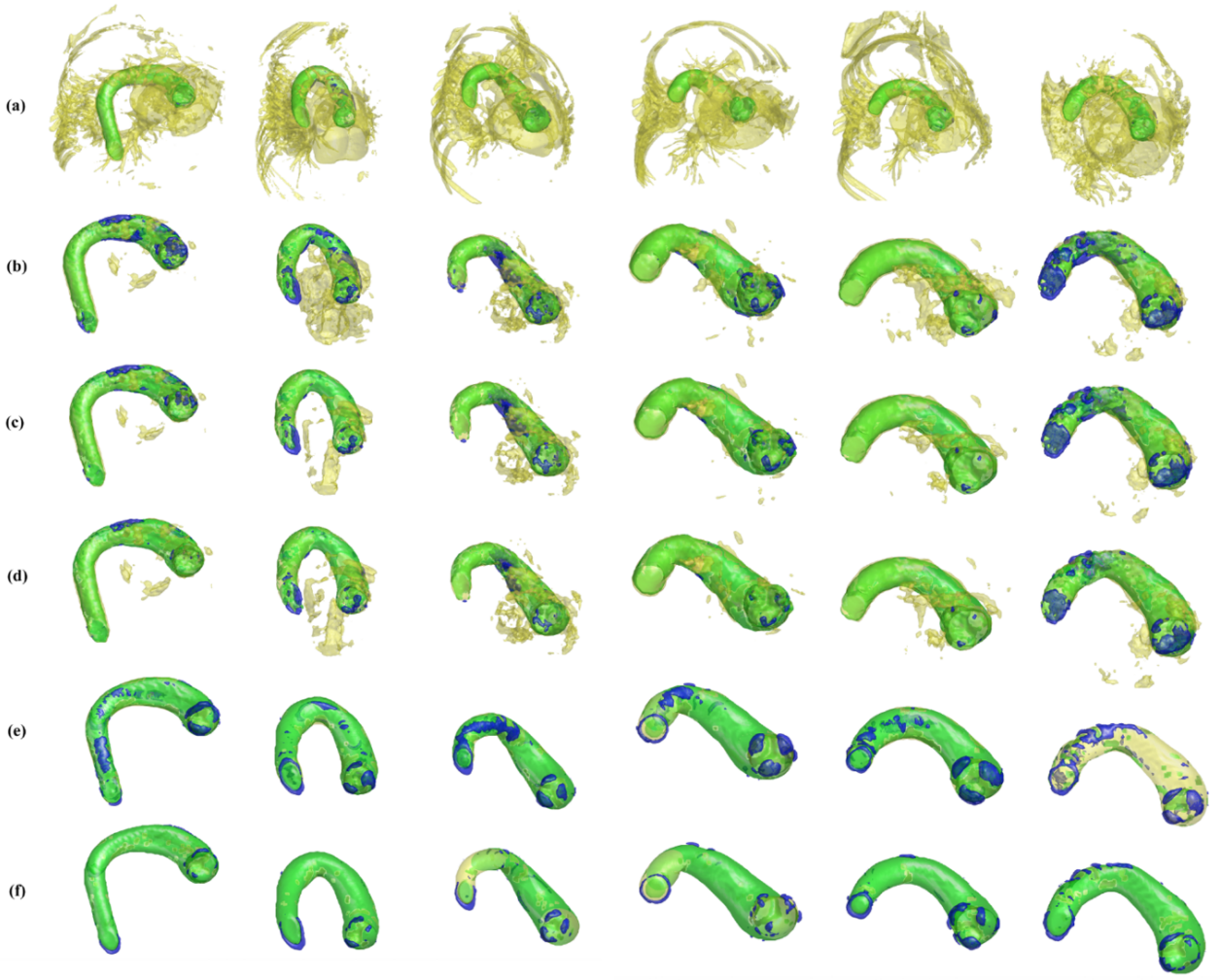


Figure 10: Qualitative results of detection and segmentation from three-fold testing at different stages on *3D CTA* dataset. (a) Naïve-Bayesian classification results. (b) Pseudo-3D CNN classification results. (c) The first round of localized interactive refining results. (d) The last round of localized interactive refining results. (e) Segmentation results using uniform-IBS. (f) Segmentation results of our approach using NU-IBS. The green, yellow, and blue colors correspond to true positive, false positive, and false negative, respectively.

Table 6: Quantitative comparison of Uniform IBS and proposed Non-Uniform IBS on *3D CTA* dataset.

#B-splines	Rate	Method	Dice	Jaccard	Similarity	Mutual Info	Hausdorff	Mahanabolis	Recall	Fallout
23	6	IBS	0.9095	0.8346	0.9173	0.1180	7.5909	0.0739	0.8406	0.0002
		NU-IBS	0.9296	0.8688	0.9314	0.1240	5.7885	0.0407	0.8702	0.0001
	3	IBS	0.9233	0.8580	0.9332	0.1221	6.9412	0.0632	0.8658	0.0002
		NU-IBS	0.9404	0.8881	0.9441	0.1279	5.8728	0.0381	0.8912	0.0001
28	6	IBS	0.9264	0.8633	0.9372	0.1230	6.7426	0.0602	0.8720	0.0002
		NU-IBS	0.9470	0.8997	0.9521	0.1300	5.8706	0.0359	0.9041	0.0001
	3	IBS	0.9304	0.8701	0.9425	0.1243	6.2202	0.0550	0.8800	0.0002
		NU-IBS	0.9536	0.9115	0.9594	0.1321	5.5733	0.0315	0.9166	0.0001

Table 7: Speed and Approximation Accuracy of proposed NU-IBS on a $256 \times 256 \times 200$ volume using single thread. (Dist Trans: Signed Distance Transformation; Chol Decomp: Cholesky Decomposition.)

#B-splines	Matrix Size	#Points	Dist Trans	Basis Matrix	Chol Decomp	Ave Error	1-Iter Deform	Interpretation
23	12167^2	62866	19.256s	12.041s	16.969s	0.020	30.253s	226.950s
28	21952^2	62866	19.704s	12.779s	48.390s	0.032	30.460s	226.973s

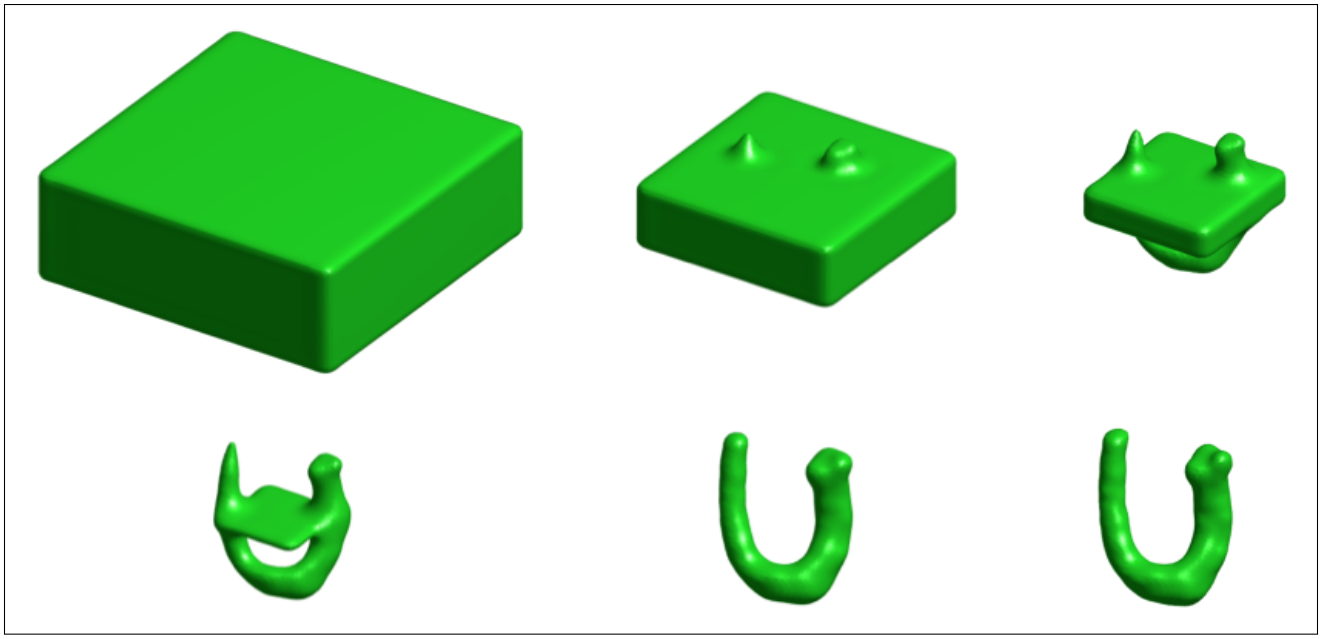


Figure 11: A example of deformation process at the 1st, 4th, 8th, 10th, 12th and 20th iterations with $\tau = 2.50e\text{-}1$ on *3D CTA* dataset.

Table 8: Quantitative Comparison on BraTS2015 Dataset

Method	Dice Coefficient (%)
DeepMedic [46]	83.87 ± 8.72
HighRes3DNet [64]	85.47 ± 8.66
DeepIGeoS [21]	**89.93 ± 6.49**
Proposed Method	89.08 ± 3.81

using *EvaluateSegmentation Tool* [61]. Table 6 shows that given the same number of B-splines and sampling rate, NU-IBS outperforms IBS in all aspects. Fig. 10 row (e) shows three qualitative segmentation results, where the false negatives in blue color can be largely observed at the tips of valves. The best performance is given when 28 B-splines and a sampling rate of every 3 pixels are used, where the highest recall rate (91.66%) and lowest Hausdorff distance (5.5733) are achieved. Although compared to the recall rate (94.83%) achieved by the interactive refinement, the region based deformation decreases by 3.17% in average, which is mainly caused by the intrinsic smoothness property of NU-IBS, where it is an inevitable issue of all PIR approaches. However, compared to the uniform IBS, the proposed NU-IBS has much richer details of subtle structures. Fig. 9 shows the qualitative comparison of uniform IBS (top row) and proposed NU-IBS (bottom row), where the uniform method tends to smooth out the geometrical details of aorta valves that are well preserved by our method. The main reason is that the IBS has far less B-spline patches at the valve regions compared to the NU-IBS, which prevents the IBS deforming further to match the data support. In Fig. 10, rows (e) and (f) show the final segmentation results using normal IBS and proposed NU-IBS respectively, where the results with less blue (false negative) and yellow (false positive) regions were obtained using our approach. Fig. 11 shows an example of region based deformation using a cube as an initialization, where the shape can break naturally during the deformation.

The proposed method was evaluated on a machine with a 3.4-GHz Intel i7 (Sandy Bridge) CPU, 32GiB of RAM, and a Nvidia GeForce Titan X (Maxwell, 12GiB GRAM) GPU. The classification speeds of Naïve-Bayesian classifier and Pseudo-3D CNN classifier are 2,725,033 voxels/second and 18,985 voxels/second on average, respectively. The NU-IBS and uniform IBS have the same computational complexity, we evaluated the speed efficiency of our method on a volume with a fixed size of $256 \times 256 \times 200$. The single thread speed and the approximation accuracies of NU-IBS are reported in Table 7, where a sampling rate of 6 pixels was used. The total computational time can be further reduced to 59s and 67s for 23 and 28 B-splines cases respectively by using multi-threading techniques and optimized factorization libraries, where *Intel TBB* [62] and *SuiteSparse* [63] were used in our case.

4.3 Segmentation Evaluation on 3D MRI Dataset

Gliomas are primary brain tumors that are a type of canceration of neuroglial cell and commonly seen in adults. MRI is an effective way to diagnose glioma, followed by necessary treatments which may include observation, surgery, radiation therapy, and chemotherapy. Various MR protocols can be used to highlight targeted regions, such as T2 and Fluid-Attenuated

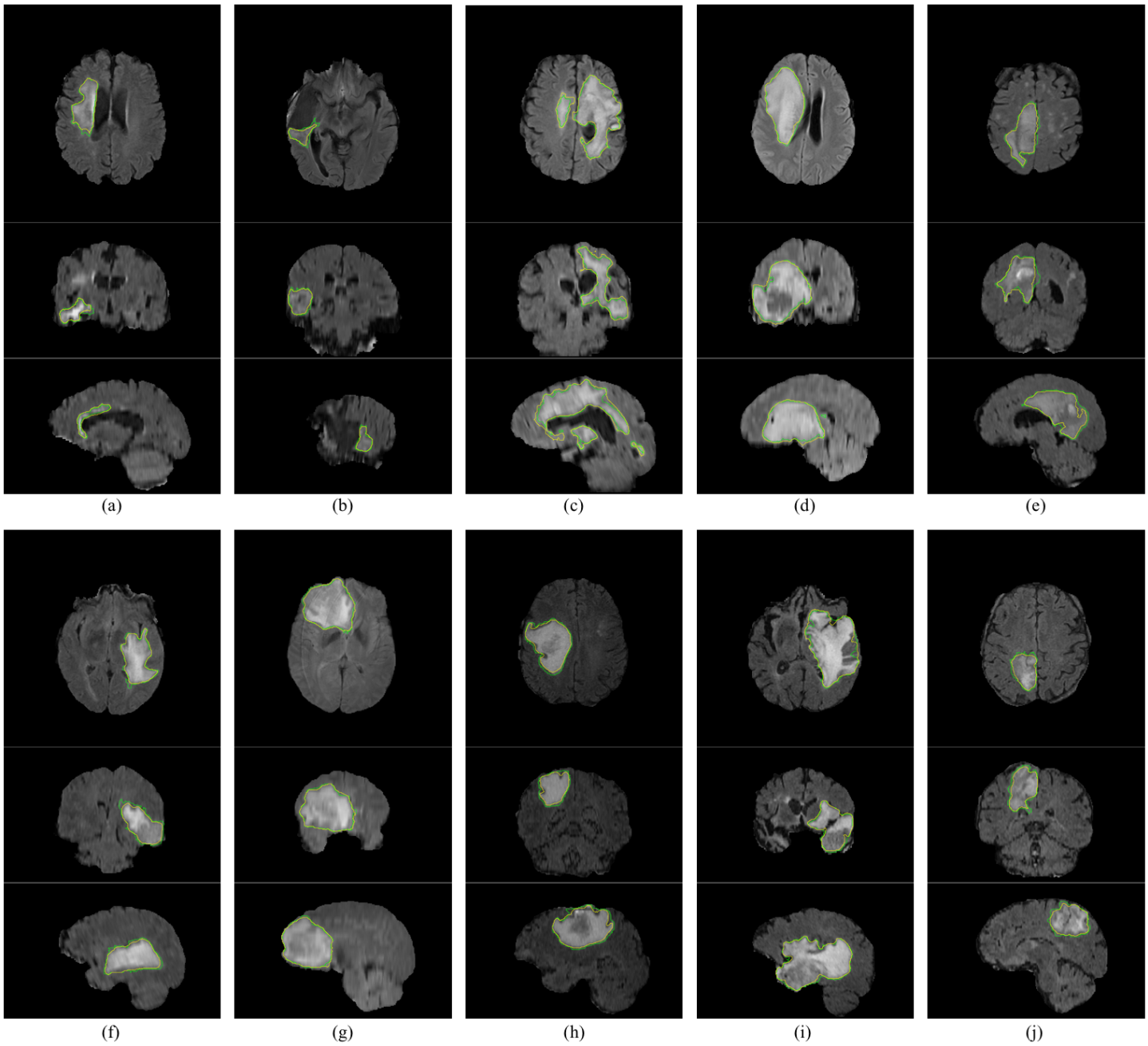


Figure 12: Examples of brain tumor segmentation on BraTS2015 dataset using our proposed method. The contours in green and yellow correspond to ground truths and our results respectively. Each group of example contains three images from axial, coronal and sagittal planes reconstructed from the 3D volume.

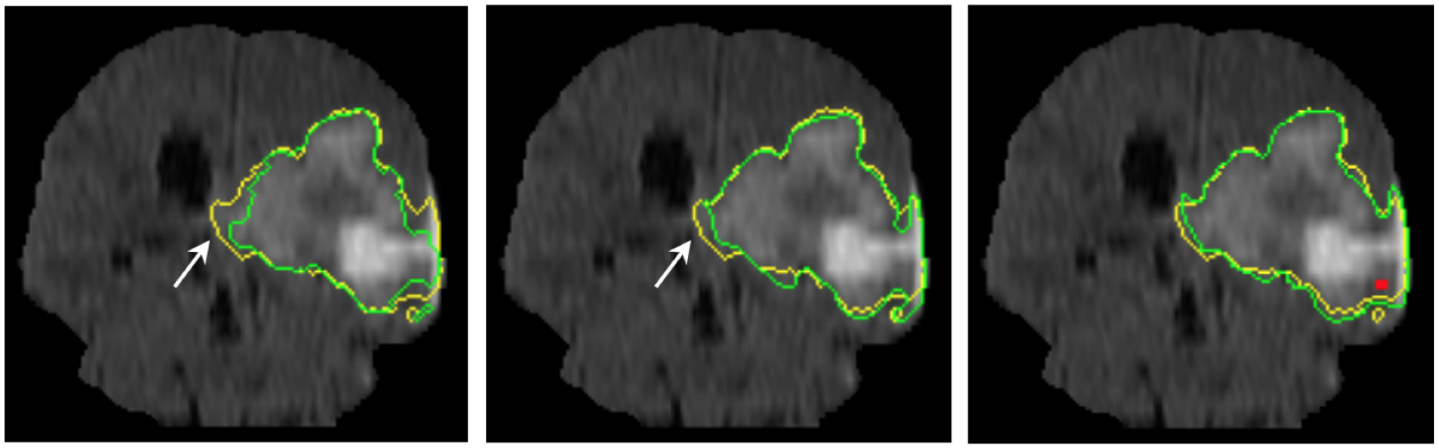


Figure 13: Visual examples of results provided in [21] for three different methods on the BraTS2015 dataset. The contours in yellow and green correspond to ground truths and segmentation, respectively.

Inversion Recovery (FLAIR) MRI highlighting differences in tissue water relaxational properties [65]. Glioma originates in glial cell and grows inside of cerebral parenchyme surrounded by other tissues in the brain, which poses great challenges to design automatic or semi-automatic segmentation methods. For example, the tumors vary significantly among individual patients in size, shape and location of the brain. In addition, the boundaries between normal tissues and lesions are often ambiguous (see Fig. 12).

BraTS2015 dataset [65] was used to evaluate the proposed 3D interactive segmentation method and to compare our method with three state-of-the-art methods [46, 64, 21]. The dataset contains 274 cases, where four types of intra-tumoral structures, namely *edema*, *non-enhancing (solid) core*, *necrotic (or fluid-filled) core*, and *non-enhancing core* were manually labeled by a trained team of radiologists and altogether seven radiographers. Following [21], 234 cases from BraTS2015 were randomly selected as training set and the remaining 40 cases were used for testing, and the FLAIR modality was used for segmentation. All volumes are skull-stripped and have a size of $240\times240\times155$ voxels with $1.0\times1.0\times1.0\text{mm}^3$ voxel spacing. The volumes were linearly scaled to $[0, 1]$ followed by a z-score normalization. We followed the same segmentation procedure as described in Sec. 4.2, with 4 additional feature aggregation blocks (see Fig. 2 (c)) for Pseudo-3D CNN due to large variation across subjects and much more ambiguous boundaries. To simulate interactive segmentation, during the refinement step, we randomly select 20 strokes (10 strokes each for background and foreground) from false positive and false negative samples in the volume, where each stroke contains a maximum of 256 voxels. The refinement will terminate when the increment of accuracy is lower than the minimum threshold (1%) or the maximum number of iterations (10) is reached.

Fig. 12 shows 10 examples of segmentation using the proposed method, where the contours in green and yellow correspond to ground truths and our results respectively. It can be observed that the proposed method is able to handle complex geometries (see Fig. 12 (c) and (i)) and blurred boundaries (see Fig. 12 (b) and (e)). It can be seen that the segmented boundaries strongly collocated with the groundtruth. The contours segmented by the proposed method are often smoother and slightly more compact than groundtruth. This is mainly caused by the smoothing effect of NU-IBS representation. Table 8 shows quantitative comparison with state-of-the-art 3D CNN segmentation methods, i.e. two fully automatic methods, *DeepMedic* [46], *HighRes3DNet* [64] and an interactive method, *DeepIGeoS* [21]. The proposed method, achieving $89.08\pm3.81\%$ in Dice coefficient, is slightly lower than *DeepIGeoS* by 0.85%, whereas our results are more consistent across all testing volumes with the lowest standard deviation (3.81% vs. 6.49% of *DeepIGeoS*). Our CNN model is also much simpler and more efficient to train. The proposed method performed significantly better than the other two full-3D CNNs. The accuracy of our method can be further improved by locally manipulating the segmented 3D surface as we demonstrated in Sec. 4.1 and Fig. 6, whereas the other methods are incapable to carry out such post-editing in 3D directly. Fig. 13 shows some examples of segmentation results obtained from these three competing methods.

By cascading two classifiers with different complexities, there are further improvements in efficiency. For BraTS2015 dataset, each volume contains 9,216,000 voxels. The Naïve-Bayesian classifier needs to evaluate all voxel position but it took on average merely 3.38 seconds, i.e. 2,725,033 voxels per second. Our CNN model only needs to be evaluated on a very small percentage of voxels due to cascading and on average it only took 1.82 (+/-0.02) seconds. Each round of refinement took 49.42 ($\pm$7.60) seconds on a Nvidia Quadro GP100 (Pascal 16GiB GRAM) GPU, which includes sampling strokes, fine-tuning model and performing prediction, and 3.4 ($\pm$2.19) rounds were carried out for each volume. However, the commonly used interactive approach [66] takes 6 minutes to cut volumes of 2-8M voxels and manual labeling will take even longer.

5 Conclusion

In this paper, we first introduced a two-stage object detection cascade that contains a fast Naïve-Bayesian classifier and a powerful Pseudo-3D CNN classifier, which balances the speed efficient and discrimination performance. Particularly, the Pseudo-3D classifier learns the hierarchical feature and decision boundary simultaneously through a supervised classification task, hence, no hand-feature-crafting is required. In addition, it avoids using computational expensive 3D convolutional operator. Localized interactive refining scheme enables user correct and refine miss-classification on the fly. Segmentation is obtained via regularizing the voxel-wise classification with manifold deformation given prediction scores. NU-IBS has non-uniform distribution of control knot that is adapted to the density of geometrical complexity, which can well preserve the subtle structures. Proposed method was evaluated on a 3D CTA dataset and BraTS2015 3D MRI dataset. The qualitative and quantitative comparisons showed the superiorities of proposed method in both segmentation accuracy and in preserving subtle structures.

References

[1] F. Zhao and X. Xie, "An overview of interactive medical image segmentation," *Annals of the BMVA*, vol. 2013, no. 7, pp. 1–22, 2013.

[2] K. McGuinness and N. E. O'connor, "A comparative evaluation of interactive segmentation algorithms," *Pattern Recognition*, vol. 43, no. 2, pp. 434–444, 2010.

[3] C. Rother, V. Kolmogorov, and A. Blake, ""grabcut" interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 309–314, 2004.

[4] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr, "Interactive image segmentation using an adaptive gmmrf model," in *European Conference on Computer Vision*. Springer, 2004, pp. 428–441.

[5] M. Unger, T. Pock, W. Trobin, D. Cremers, and H. Bischof, "Tvseg-interactive total variation based image segmentation." in *British Machine Vision Conference*, vol. 31. Citeseer, 2008, pp. 44–46.

[6] J. Santner, M. Unger, T. Pock, C. Leistner, A. Saffari, and H. Bischof, "Interactive texture segmentation using random forests and total variation." in *British Machine Vision Conference*, 2009, pp. 1–12.

[7] M. Jian and C. Jung, "Interactive image segmentation using adaptive constraint propagation," *IEEE Transactions on Image Processing*, vol. 25, no. 3, pp. 1301–1311, 2016.

[8] J.-L. Jones, X. Xie, and E. Essa, "Combining region-based and imprecise boundary-based cues for interactive medical image segmentation," *International Journal for Numerical Methods in Biomedical Engineering*, vol. 30, no. 12, pp. 1649–1666, 2014.

[9] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in nd images," in *International Conference on Computer Vision*, vol. 1. IEEE, 2001, pp. 105–112.

[10] 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.

[11] S. Han, W. Tao, D. Wang, X.-C. Tai, and X. Wu, "Image segmentation based on grabcut framework integrating multiscale nonlinear structure tensor," *IEEE Transactions on Image Processing*, vol. 18, no. 10, pp. 2289–2302, 2009.

[12] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman, "Geodesic star convexity for interactive image segmentation," in *IEEE conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 3129–3136.

[13] B. L. Price, B. Morse, and S. Cohen, "Geodesic graph cut for interactive image segmentation," in *IEEE conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 3161–3168.

[14] S. Meena, V. S. Prasath, Y. M. Kassim, R. J. Maude, O. V. Glinskii, V. V. Glinsky, V. H. Huxley, and K. Palaniappan, "Multiquadric spline-based interactive segmentation of vascular networks," in *IEEE Engineering in Medicine and Biology Society*. IEEE, 2016, pp. 5913–5916.

[15] S. Meena, V. B. S. Prasath, K. Palaniappan, and G. Seetharaman, "Elastic body spline based image segmentation," in *International Conference on Image Processing*. IEEE, 2014, pp. 4378–4382.

[16] X. Dong, J. Shen, L. Shao, and M.-H. Yang, "Interactive cosegmentation using global and local energy optimization." *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3966–3977, 2015.

[17] J. Feng, B. Price, S. Cohen, and S.-F. Chang, "Interactive segmentation on RGBD images via cue selection," in *IEEE conference on Computer Vision and Pattern Recognition*, 2016, pp. 156–164.

[18] Y. Lu, X. Bai, L. Shapiro, and J. Wang, "Coherent parametric contours for interactive video object segmentation," in *IEEE conference on Computer Vision and Pattern Recognition)*, 2016, pp. 642–650.

[19] H. Isack, O. Veksler, M. Sonka, and Y. Boykov, "Hedgehog shape priors for multi-object segmentation," in *IEEE conference on Computer Vision and Pattern Recognition*, 2016, pp. 2434–2442.

[20] N. Xu, B. Price, S. Cohen, J. Yang, and T. S. Huang, "Deep interactive object selection," in *IEEE conference on Computer Vision and Pattern Recognition*, 2016, pp. 373–381.

[21] G. Wang, M. A. Zuluaga, W. Li, R. Pratt, P. A. Patel, M. Aertsen, T. Doel, A. L. Divid, J. Deprest, S. Ourselin *et al.*, "DeepIGeoS: a deep interactive geodesic framework for medical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[22] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, pp. 504–507, 2006.

[23] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[24] M. Lai, "Deep learning for medical image segmentation," *arXiv preprint arXiv:1505.02000*, 2015.

[25] B. S. Morse, W. Liu, T. S. Yoo, and K. Subramanian, "Active contours using a constraint-based implicit representation," in *IEEE conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2005, pp. 285–292.

[26] X. Xie and M. Mirmehdi, "Radial basis function based level set interpolation and evolution for deformable modelling," *Image and Vision Computing*, vol. 29, no. 2, pp. 167–177, 2011.

[27] A. Paiement, M. Mirmehdi, X. Xie, and M. C. Hamilton, "Integrated segmentation and interpolation of sparse data," *IEEE Transactions on Image Processing*, vol. 23, no. 1, 2014.

[28] A. Gelas, O. Bernard, D. Friboulet, and R. Prost, "Compactly supported radial basis functions based collocation method for level-set evolution in image segmentation," *IEEE Transactions on Image Processing*, vol. 16, no. 7, pp. 1873–1887, 2007.

[29] T. Sahin and M. Unel, "Fitting globally stabilized algebraic surfaces to range data," in *International Conference on Computer Vision*, vol. 2. IEEE, 2005, pp. 1083–1088.

[30] O. Bernard, D. Friboulet, P. Thévenaz, and M. Unser, "Variational b-spline level-set: a linear filtering approach for fast deformable model evolution," *IEEE Transactions on Image Processing*, vol. 18, no. 6, pp. 1179–1191, 2009.

[31] M. Rouhani and A. D. Sappa, "The richer representation the better registration," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 5036–5049, 2013.

[32] M. Rouhani, A. D. Sappa, and E. Boyer, "Implicit b-spline surface reconstruction," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 22–32, 2015.

[33] X. Xie and M. Mirmehdi, "MAC: Magnetostatic active contour model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 4, pp. 632–646, 2008.

[34] A. Dubrovina-Karni, G. Rosman, and R. Kimmel, "Multi-region active contours with a single level set function," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 8, pp. 1585–1601, 2015.

[35] R. Delgado-Gonzalo, D. Schmitter, V. Uhlmann, and M. Unser, "Efficient shape priors for spline-based snakes." *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3915–3926, 2015.

[36] A. Badoual, D. Schmitter, V. Uhlmann, and M. Unser, "Multiresolution subdivision snakes," *IEEE Transactions on Image Processing*, vol. 26, no. 3, pp. 1188–1201, 2017.

[37] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[38] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 359–369, 1998.

[39] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International Journal of Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997.

[40] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Shape modeling with front propagation: A level set approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 158–175, 1995.

[41] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations," *Journal of Computational Physics*, vol. 79, no. 1, pp. 12–49, 1988.

[42] J. A. Sethian, *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science.* Cambridge University Press, 1999, vol. 3.

[43] S. Osher and R. Fedkiw, *Level set methods and dynamic implicit surfaces.* Springer, 2006.

[44] 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.

[45] M. Botsch, D. Bommes, and L. Kobbelt, "Efficient linear system solvers for mesh processing," in *Mathematics of Surfaces XI.* Springer, 2005, pp. 62–83.

[46] K. Kamnitsas, C. Ledig, V. F. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker, "Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation," *Medical Image Analysis*, vol. 36, pp. 61–78, 2017.

[47] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[48] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *IEEE conference on Computer Vision and Pattern Recognition*, 2016, pp. 761–769.

[49] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," in *IEEE conference on Computer Vision and Pattern Recognition*, 2019, pp. 558–567.

[50] T. A. Davis, *Direct methods for sparse linear systems.* Society for Industrial and Applied Mathematics, 2006.

[51] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger, "Robust anisotropic diffusion," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 421–432, 1998.

[52] T. Chan, A. Marquina, and P. Mulet, "High-order total variation-based image restoration," *SIAM Journal on Scientific Computing*, vol. 22, no. 2, pp. 503–516, 2000.

[53] L. A. Vese and T. F. Chan, "A multiphase level set framework for image segmentation using the mumford and shah model," *International Journal of Computer Vision*, vol. 50, no. 3, pp. 271–293, 2002.

[54] 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 Transactions on Medical Imaging*, vol. 20, no. 1, pp. 45–57, 2001.

[55] T. F. Chan, S. Esedoglu, and M. Nikolova, "Algorithms for finding global minimizers of image segmentation and denoising models," *SIAM Journal on Applied Mathematics*, vol. 66, no. 5, pp. 1632–1648, 2006.

[56] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.

[57] H.-K. Zhao, T. Chan, B. Merriman, and S. Osher, "A variational level set approach to multiphase motion," *Journal of Computational Physics*, vol. 127, pp. 179–195, 1996.

[58] G. Turk and M. Levoy, "Zippered polygon meshes from range images," in *SIGGRAPH The Annual conference on Computer Graphics and Interactive Techniques.* ACM, 1994, pp. 311–318.

[59] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *SIGGRAPH The Annual conference on Computer Graphics and Interactive Techniques.* ACM, 1996, pp. 303–312.

[60] 3Dim Laboratory s.r.o., "3DimViewer," http://www.3dim-laboratory.cz/software/3dimviewer, accessed: 2017-03-01.

[61] A. A. Taha and A. Hanbury, "Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool," *BMC MI*, vol. 15, p. 29, August 2015.

[62] Intel Corporation, "Intel Threading Building Blocks," https://www.threadingbuildingblocks.org/, accessed: 2017-03-01.

[63] T. Davis, "SUITESPARSE," http://faculty.cse.tamu.edu/davis/suitesparse, accessed: 2017-03-01.

[64] W. Li, G. Wang, L. Fidon, S. Ourselin, M. J. Cardoso, and T. Vercauteren, "On the compactness, efficiency, and representation of 3d convolutional networks: brain parcellation as a pretext task," in *International Conference on Information Processing in Medical Imaging*. Springer, 2017, pp. 348–360.

[65] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest *et al.*, "The multimodal brain tumor image segmentation benchmark (brats)," *IEEE Transactions on Medical Imaging*, vol. 34, no. 10, pp. 1993–2024, 2014.

[66] K. Li, X. Wu, D. Z. Chen, and M. Sonka, "Optimal surface segmentation in volumetric images-a graph-theoretic approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 119–134, 2005.