3-D Surface Segmentation Meets Conditional Random Fields

Leixin Zhou, Zisha Zhong, Abhay Shah and Xiaodong Wu, *Senior Member, IEEE*

***Abstract*—Automated surface segmentation is important and challenging in many medical image analysis applications. Recent deep learning based methods have been developed for various object segmentation tasks. Most of them are a classification based approach, e.g. U-net, which predicts the probability of being target object or background for each voxel. One problem of those methods is lacking of topology guarantee for segmented objects, and usually post processing is needed to infer the boundary surface of the object. In this paper, a novel model based on 3-D convolutional neural network (*CNN*) and Conditional Random Fields (*CRFs*) is proposed to tackle the surface segmentation problem with end-to-end training. To the best of our knowledge, this is the first study to apply a 3-D neural network with a *CRFs* model for direct surface segmentation. Experiments carried out on NCI-ISBI 2013 MR prostate dataset and Medical Segmentation Decathlon Spleen dataset demonstrated very promising segmentation results.**

***Index Terms*—Surface segmentation, deep learning, CNN, CRFs, shape prior, 3-D.**

I. Introduction

AUTOMATED Image segmentation plays a very import role in quantitative image analysis. In recent years, semantic segmentation methods based on convolutional neural networks (*CNNs*) become very popular in computer vision and then in medical imaging research communities, e.g. the fully convolutional networks (FCNs) [1] with application in natural images, and then U-net [2] and its 3-D version V-net [3] for medical image segmentation.

In natural and medical images, as pixels or voxels usually exhibit strong correlation, jointly modeling the label distribution globally and/or locally is desirable. To capture the contextual information, conditional random fields (*CRFs*) [4] are commonly utilized for semantic segmentation. The model consists of a unary potential term and a pairwise potential term. The unary potential specifies the per-pixel or voxel confidence of assigning a label, while the pairwise potential regularizes the label smoothness between neighboring voxels. In computer vision, the *CRFs* model has been integrated with *CNNs* for an end-to-end training to take advantages of both the modeling power of *CRFs* and the representation-learning ability of *CNNs* [5].

Most deep learning based semantic segmentation methods are *region-based* [1], [2], [3], [5], in which each pixel is labeled as either target object or background. On the other hand, one can also formulate semantic segmentation with a surface based model, in which the boundary surface of the the target object is computed directly. Apparently these two types of approaches are equivalent as the boundary surface can be computed from the labeled target volume, and vice versa. As one of prominent surface-based methods, Graph-Search (GS) [6], [7] has achieved great success, especially in medical imaging field, e.g. [8], [9], [10], [11], [12]. This method is capable of simultaneously detecting multiple interacting surfaces with global optimality with respect to the energy function designed for individual surfaces with several geometric constraints defining the surface smoothness and interrelations. The method solves the surface segmentation problem by transforming it into computing a minimum *s-t* cut in a derived arc-weighted directed graph, which can be solved with global optimality and has a low-order polynomial time complexity.

Inspired by the graph search method, Shah *et al.* [13], [14] first modeled the terrain-like surfaces segmentation as direct surface identification using a regression network based on *CNN*. The network only models the unary potentials. As the prediction was directly on surface positions, a surface monotonicity constraint was realized in a straightforward way. The network used was a very light weighted 2-D *CNN* and no post processing was required. Surprisingly the results were very promising.

It would be of high interest to extend Shah *et al.*'s method to 3-D for segmenting general non-terrain like surface. To achieve that goal, two major obstacles need to be overcome: 1) how to generate patches with a regular neighborhood in 3-D, such that the traditional *CNN* can be applied? 2) It is generally hard to train a 3-D network, especially when it contains giant fully connected (*FC*) layers, the size of which is closely related to inference/patch size. There is a tradeoff between the amount of contextual information within a patch and the number of parameters in a network architecture, i.e. a bigger patch size comes along with more contextual information, but more parameters need to be trained.

Contributions: To overcome those technical barriers, we propose to build a framework of *surface-based CNN+CRFs* for surface segmentation in medical images. The contributions are twofold: 1) A *surface-based CNN+CRFs* framework is proposed, including proper modeling of unary and pairwise term, and compatibility matrix customized for surface segmentation; 2) A novel shape-aware patch generation method, which is based on harmonic mapping, is also proposed to make efficient training of surface segmentation possible.

L. Zhou, Z. Zhong, A. Shah and X. Wu are with the Department of Electrical and Computer Engineering, The University of Iowa, Iowa City, IA, 52242 USA (e-mail: leixin-zhou@uiowa.edu, zisha-zhong@uiowa.edu, abhay-shah-1@uiowa.edu, xiaodong-wu@uiowa.edu).

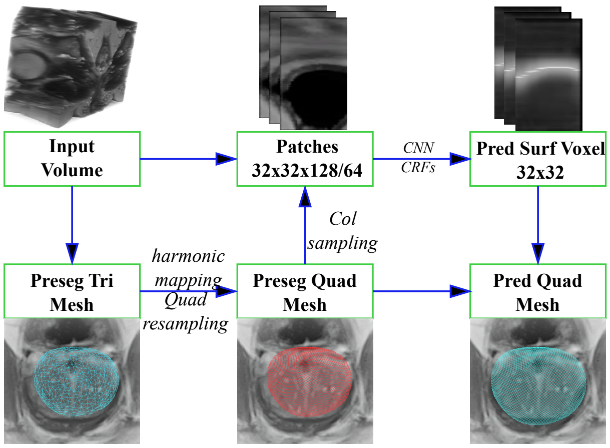

Fig. 1. Overall flowchart of the proposed method. With some pre-segmentation method, a coarse surface segmentation or pre-segmentation (Preseg) can be generated and represented as a triangular (Tri) mesh. Then the proposed remeshing method would convert the Preseg Tri mesh into a Preseg quadrilateral (Quad) mesh. Based on the Preseg Quad mesh, for each vertex in which, sampling within input volume in it's normal direction with fixed resolution and length would generate a column. Combining the columns for all vertices on the Preseg Quad mesh produces the 3-D volumes/patches, which are the inputs for the proposed network. Then the network predicts surface voxels, which finally gives the prediction (Pred) Quad mesh surface.

II. Method

The pipeline starts with a pre-segmentation (Preseg), which serves as the *coarse* surface position and *topology* that the final segmentation should comply with. The *triangular* (Tri) mesh of the pre-segmented surface is then converted to a *quadrilateral* (Quad) mesh, which is friendly to convolution operations. Based on the Quad mesh, image patches, which contain terrain-like boundary surfaces of the partial target object can be generated and are fed into the proposed neural network to predict the voxels on the desired surface. The flowchart of proposed method is illustrated in Fig. 1.

In the following sections, we will first define the *surface-based* segmentation problem rigorously. Then the *CRFs* model will be reviewed, and then we will cover the modeling of the unary and pairwise terms. A novel shape-aware patch generation method and the network architecture will be explained finally.

A. Surface-Based Segmentation

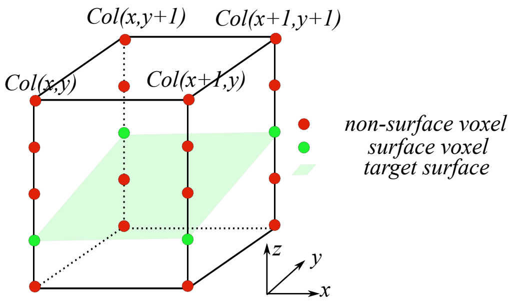

Fig. 2. Definition of the surface segmentation for volumetric images.

A 3-D image can be viewed as a 3-D tensor $\mathcal{I}(x, y, z)$. A terrain-like *surface* in $\mathcal{I}$ is oriented and shown in Fig. 2. Let X, Y and Z denote the image sizes in x, y and z dimensions, respectively. The surface is defined by a function $\mathcal{N} : (x, y) \to \mathcal{N}_{x,y}$, where $x \in \{0, \dots, X-1\}$, $y \in \{0, \dots, Y-1\}$, and $\mathcal{N}_{x,y} \in \{0, \dots, Z-1\}$. Thus any surface in $\mathcal{I}$ intersects with exactly one voxel of each *column* (*Col*) in parallel with z direction, and it consists of exactly $X \times Y$ voxels. In surface segmentation, generally we define the energy or cost for one feasible surface $\mathcal{N}$ to be:

$$E(\mathcal{N}|\mathcal{I}) = w_u E_u(\mathcal{N}|\mathcal{I}) + (1 - w_u) E_p(\mathcal{N}|\mathcal{I}), \qquad (1)$$

and the "optimal" surface is computed by minimizing the energy. Generally the unary term E_u is the energy when considering each *column* independently, and the pairwise energy term E_p penalizes discontinuity of surface position among adjacent *columns*, and w_u is to balance the contributions of the two terms.

B. CRFs Model

In this section the Conditional Random Fields (*CRFs*) model is first briefly reviewed and then the modeling of unary and pairwise terms is explained.

1) Review: CRFs is defined on observations $\mathcal{X}$ and random variables $\mathcal{Y}$ as follows: Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph such that $\mathcal{Y} = (\mathcal{Y}_v)_{v \in \mathcal{V}}$, so that $\mathcal{Y}$ is indexed by the vertices of $\mathcal{G}$. Then $(\mathcal{X}, \mathcal{Y})$ is a conditional random field when the random variables $\mathcal{Y}_v$ conditioned on $\mathcal{X}$, obey the markov property.

The pair $(\mathcal{X}, \mathcal{Y})$ can be characterized by a Gibbs distribution of the form

$$P(\mathcal{Y} = \boldsymbol{y}|\mathcal{X}) = \frac{1}{\mathcal{Z}(\mathcal{X})} \exp(-E(\boldsymbol{y}|\mathcal{X})). \qquad (2)$$

Here $E(\boldsymbol{y}|\mathcal{X})$ is called the energy of the configuration $\boldsymbol{y}$ and $\mathcal{Z}(\mathcal{X})$ is the partition function. For convenience, the conditioning on $\mathcal{X}$ is dropped. In the fully connected pairwise *CRFs* model of [15], the energy of a label assignment $\boldsymbol{y}$ is given by:

$$E(\boldsymbol{y}) = w_u \sum_{v} \psi_u(\boldsymbol{y}_v) + (1 - w_u) \sum_{(v,v') \in \mathcal{E}} \psi_p(\boldsymbol{y}_v, \boldsymbol{y}_{v'}), \quad (3)$$

where the unary energy components $\psi_u(\boldsymbol{y}_v)$ measure the energy of $\mathcal{Y}_v$ taking the value $\boldsymbol{y}_v$, and pairwise energy components $\psi_p(\boldsymbol{y}_v, \boldsymbol{y}_{v'})$ measure the energy of assigning $\boldsymbol{y}_v$, $\boldsymbol{y}_{v'}$ to $\mathcal{Y}_v$ and $\mathcal{Y}_{v'}$ simultaneously. w_u balances the two terms.

C. Modeling the Surface Segmentation as CRFs

It should be natural to model the *surface-based* segmentation with a *CRFs* model. In the *surface-based* segmentation scenario, $\mathcal{N}$ are the random variables and $\mathcal{I}$ is the observation. The unary potentials $\psi_u(n_{x,y})$ correspond to the energy of assigning the surface position to be $n_{x,y}$ without explicitly considering the column interactions. And the pairwise potentials $\psi_p(n_{x,y}, n_{x',y'})$ represent the energy to simultaneously assign surface positions to be $n_{x,y}$ and $n_{x',y'}$, respectively.

1) Modeling Unary and Pairwise Potentials: Commonly, unary energies are obtained from a *CNN*, which, roughly speaking, predicts labels for *columns* without explicitly considering the smoothness and the consistency of label assignments. The pairwise energies provide the smoothing term that encourages assigning similar labels to *columns* with similar properties. In [15], the pairwise potentials are modeled as weighted Gaussians:

$$\begin{aligned}\psi_p(n_{x,y}, n_{x',y'}) &= \mu(n_{x,y}, n_{x',y'})k(\mathrm{f}_{x,y}, \mathrm{f}_{x',y'})\\ &= \mu(n_{x,y}, n_{x',y'}) \sum_{m=1}^{M} w^{(m)} k_G^m(\mathrm{f}_{x,y}, \mathrm{f}_{x',y'}),\end{aligned} \tag{4}$$

where each k_G^m for $m = 1, ..., M$, is a Gaussian kernel applied on feature vectors $\mathrm{f}_{x,y}$ and $\mathrm{f}_{x',y'}$. The feature vectors of $Col_{x,y}$, denoted by $\mathrm{f}_{x,y}$, are derived from image features such as spatial location, and visual features like pixel/voxel intensities. The function μ, called the label compatibility function, captures the compatibility between different pairs of labels.

In [15], the term $k(\mathrm{f}_{x,y}, \mathrm{f}_{x',y'})$ is defined as:

$$\begin{aligned}k(\mathrm{f}_{x,y}, \mathrm{f}_{x',y'}) = w^{(1)}\exp\Bigg(&-\frac{||(x'-x, y'-y)||^2}{2\theta_\alpha^2}\\ &-\frac{||Col_{x,y} - Col_{x',y'}||^2}{2\theta_\beta^2}\Bigg)\\ +w^{(2)}\exp\Bigg(&-\frac{||(x'-x, y'-y)||^2}{2\theta_\gamma^2}\Bigg),\end{aligned} \tag{5}$$

where the first and second term on RHS is called *appearance kernel* and *smoothness kernel*, respectively. θ_α, θ_β, and θ_γ control shapes of corresponding Gaussian kernels.

2) Customized Pairwise Terms: Next we will explain the *CRFs* pairwise term modeling in the proposed setting.

a) Customized Visual Feature: One main difference from the common semantic segmentation is the meaning of $Col_{x,y}$. In 2-D *region-based* segmentation, $Col_{x,y}$ just reduces to 1-D (gray images) or 3-D (color images) pixel intensity values . In this sense, it is reasonable that larger difference may indicate possible label differences. However, in our *surface-based* segmentation setting, it represents one *column* of voxels in our 3-D image $\mathcal{I}$. One should notice that currently it is not proper to use term $||Col_{x,y} - Col_{x',y'}||^2$ as a measure indicating possible labeling differences for corresponding Col pairs. The reason is that the $Col_{x,y}$ may also contain voxels that are not significantly related to the labeling of the current *column* and may have large variance, e.g. the two voxels $Col_{x,y}[0]$ and $Col_{x',y'}[0]$, which are outlined by blue dash ovals in Fig. 3(b). To remedy this problem, we propose to use the *probability-map* or *logits* output by *CNN* as the visual feature. The observation is that a *CNN* with enough receptive field can in some sense get rid of these unrelated voxels in *probability-map*, which is illustrated in Fig. 3(c). The proposed kernel term is of the new form

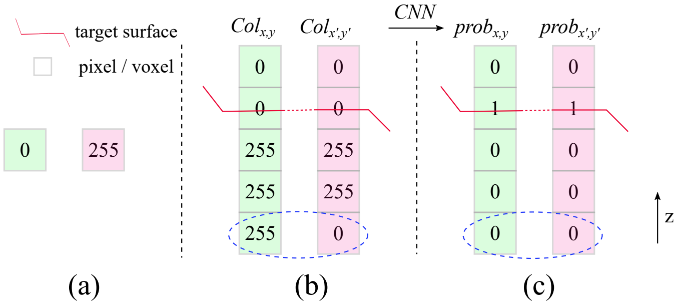

Fig. 3. Visual Features in different settings. For convenience, gray intensity value is shown. (a) In 2-D *region-based* segmentation, $Col_{x,y}$ is just a scalar intensity value. (b) In 3-D *surface-based* segmentation, $Col_{x,y}$ is a column of voxels in the z dimension. It can be noticed that within columns, intensity differences of voxel pairs that are not around the target surface actually does not relate much to the labellings of the column pairs. One voxel pair is outlined by the blue dash oval. (c) From our observation, the probability map or logits of each column is more proper to use as visual features, as the *CNN* may remove the interference of unrelated voxel pairs by having a global view.

$$\begin{aligned}k(\mathrm{f}_{x,y}, \mathrm{f}_{x',y'}) = w^{(1)}\exp\Bigg(&-\frac{||(x'-x, y'-y)||^2}{2\theta_\alpha^2}\\ &-\frac{||prob_{x,y} - prob_{x',y'}||^2}{2\theta_\beta^2}\Bigg)\\ +w^{(2)}\exp\Bigg(&-\frac{||(x'-x, y'-y)||^2}{2\theta_\gamma^2}\Bigg).\end{aligned} \tag{6}$$

b) Customized Compatibility Matrix: The other difference is the physical meaning of label differences of different *columns*. In 2-D *region-based* segmentation, the quantity of difference among different classes does not have much meaning within it. Suppose classes *cat*, *car* and *building* have labels 0, 1, and 2, respectively. Generally there is no way to give any reasonable meaning to the label difference. For example, one can not say *cat* is more compatible to *car* than to *building*. However, in our 3-D *surface-based* segmentation, the label difference has an explicit meaning of *surface smoothness*, i.e. the smaller label difference, the smoother the surface.

The naive way to learn the compatibility matrix would need to learn a $Z \times Z$ matrix. In our scenario, this way will be ill-posed, since some label pairs may not exist or at least are very rare in the training data. To tackle this, we proposed to parameterize the compatibility matrix ($Z \times Z$) with a parameter function $\mathcal{C}$. The parameterization has the following formula:

$$\begin{aligned}\mu(n_{x,y}, n_{x',y'}) &= \mathcal{C}\left(||n_{x,y} - n_{x',y'}||^1\right)\\ &= -\exp\left(-\frac{||n_{x,y} - n_{x',y'}||^2}{\theta_{\text{comp}}^2}\right).\end{aligned} \tag{7}$$

The intuition behind is that the compatibility penalty is monotonically related to the label difference. In this way, the training parameter number for compatibility matrix is reduced from $Z \times Z$ to 1.

D. Shape-Aware Patch Generation

For real applications, two obstacles need to be solved first such that we can model the *surface-based* segmentation as

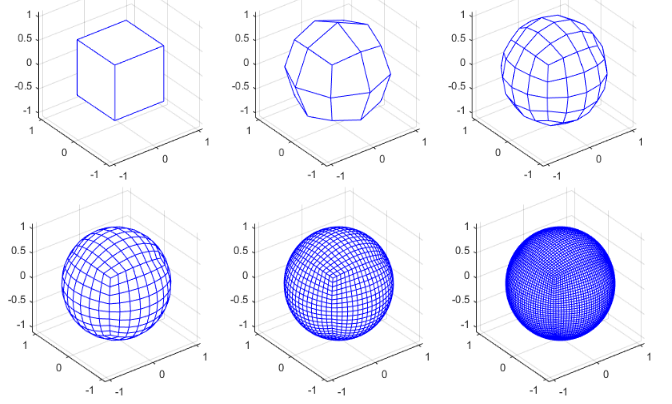

Fig. 4. Quad parameterization of the unit sphere S^{us}_{Quad} with recursion number 0 to 5, shown from the left to the right and the top to the bottom.

terrain-like surfaces segmentation using *CNN*. 1) Unfolding the surface into a terrain-like surface, on which our *surface-based* segmentation is defined. 2) The unfolded image or patche volumes should have a rectangular cuboid grid structure in 3-D, such that the traditional *CNN* can apply to.

1) To Terrain-like Surfaces: For the cylindrical surface, it is very trivial to use a cylindrical coordinate transform. For the simplest closed surface, i.e. a sphere, it is also trivial to utilize a pre-defined sampling and dividing pattern on the sphere to unfold it. But *how to deal with more complex closed surface, e.g. non-convex objects/surfaces*? We propose to tackle 1) by first harmonic mapping the surface to the unit sphere and then using a pre-defined sampling and dividing pattern to realize the unfolding.

2) To Grid Structure Patch: Given a pre-segmented surface, a *triangular* (Tri) mesh S^{ps}_{Tri} of the pre-segmentation can be computed by the marching cube method. However, the Tri mesh may contain variable neighborhood schemes and does not have a regular grid structure. We propose to resample the Tri mesh of the pre-segmentation into a *quadrilateral* (Quad) mesh to tackle 2).

The detailed explanation of the proposed pipeline for the proposed shape-aware patch generation is as follows.

a) Harmonic Mapping: The triangulated pre-segmentation S^{ps}_{Tri} mesh, which should be a Genus-0 closed surface (otherwise, it needs to make it closed artificially), is *harmonically mapped* to the unit sphere to obtain a triangulated sphere S^{mps}_{Tri} using the algorithm in [16].

b) Quad Parameterization of the Unit Sphere: The unit sphere can be parameterized by a Quad mesh (except 8 grid points, which only had 3 neighbors), denoted by S^{us}_{Quad}. This parameterization proceeds in a recursive way. The base Quad mesh $S^{us_0}_{\text{Quad}}$ is a inscribed cube of the unit sphere. In each recursion, the middle point of each edge is computed and moved outwards exactly to the unit sphere. This process is demonstrated in Fig. 4. It can be noticed that more recursive iterations produce higher grid resolution on the surface. In our experiments, the recursion number is chosen as 5. In other words, for each face of the base Quad mesh, the Quad grid size increases from the base 2×2 to $(2^5 + 1) \times (2^5 + 1)$.

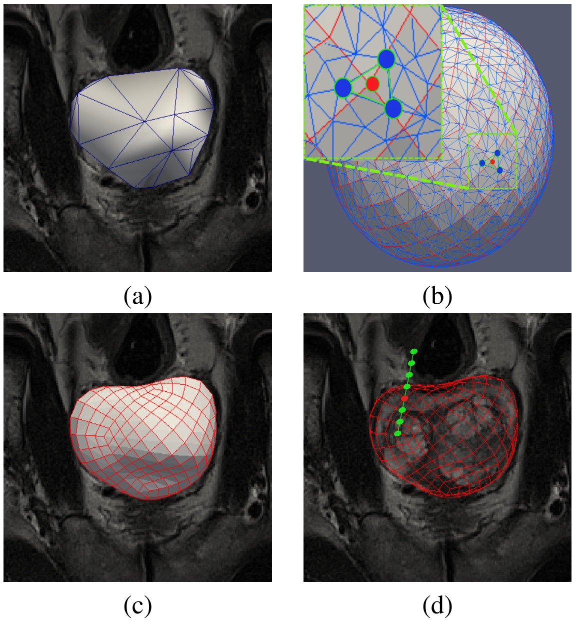

Fig. 5. The proposed process of patch generation. (a) Triangular pre-segmentation mesh S^{ps}_{Tri} (blue). Downsampling was done for demonstration. (b) Overlay of the harmonically mapped pre-segmentation surface S^{mps}_{Tri} (blue) and the Quad parameterized unit sphere S^{us}_{Quad} (red). For the red point P^{us}_{Quad} on S^{us}_{Quad}, the corresponding face F^{mps}_{Tri} is demonstrated as the triangle with three vertices colored as blue and edges colored as green. (c) Quad pre-segmentation mesh surface S^{ps}_{Quad}. (d) Sampling a column in normal direction for each P^{ps}_{Quad} as its feature. The superscripts ps, mps and us denotes the pre-segmentation, the mapped pre-segmentation and the unit sphere, respectively.

c) Quad Remeshing of Preseg Surfaces: Both the mapped pre-segmentation surface S^{mps}_{Tri} and the Quad parameterized sphere S^{us}_{Quad} are a unit sphere manifold in the same space, and can be overlaid to each other. For each grid point P^{us}_{Quad} on the Quad mesh sphere S^{us}_{Quad}, the corresponding triangular face F^{mps}_{Tri} in the mapped pre-segmentation surface S^{mps}_{Tri} can be found. The Barycentric coordinates of this grid point P^{us}_{Quad} can then be computed. For each vertex P^{ps}_{Tri} on S^{ps}_{Tri} and each P^{mps}_{Tri} on S^{mps}_{Tri}, there exist a one-to-one mapping relation. Using the Barycentric coordinates computed on the unit sphere manifold, for each P^{us}_{Quad}, we can get its approximate corresponding point P^{ps}_{Quad} on the original triangulated pre-segmentation surface S^{ps}_{Tri}. The normal direction for each P^{ps}_{Quad} can be computed in a similar fashion. In this way, a Quad remeshing for the triangulated pre-segmentation surface could be realized, denoted by S^{ps}_{Quad}. This Quad remeshing process works for all genus-0 closed surfaces, which is illustrated in Fig. 5 (b-c).

d) Sampling Columns to Generate Patches: After the remeshing, for each P^{ps}_{Quad}, we sample a column of voxels with certain length and resolution in the normal direction, which is treated as the image feature for this vertex and corresponds to one column in our problem definition (Fig. 5 (d)). The Quad surface mesh is a 2-D manifold, hence, after extending in the image feature column dimension, a 3-D volume, corresponding to $\mathcal{I}$, is generated. In addition, our Quad parameterization of a unit sphere can be easily split into 6 pieces, which correspond to the 6 faces of the inscribed cube. If we split these 6 pieces and treat the image feature column as the third dimension, 6

3-D volumes / patches can be generated.

e) Ground Truth Generation: When the S^{ps}_{Quad} is derived, the truth surface voxel for each P^{ps}_{Quad} or column can be defined as the nearest neighbor voxel to the manual segmentation mesh S_{truth} in the normal direction.

E. Network Architecture

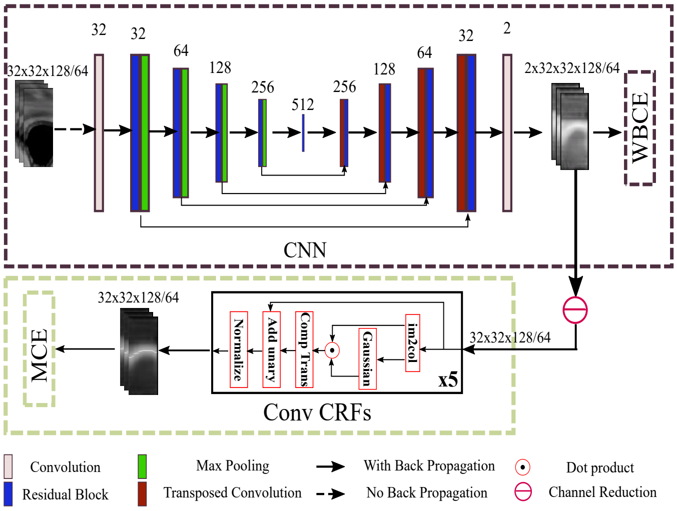

Fig. 6. The proposed surface segmentation network. It consists of the *proposed CNN* and the *proposed CRFs* layer, from the top to the bottom. For convolution and residual blocks, the number indicates dimensions of feature maps, and $3 \times 3 \times 3$ convolution kernels are utilized except the last convolution layer, where $1 \times 1 \times 1$ kernels are chosen; for pooling and transposed convolution layers, $2 \times 2 \times 2$ kernels and strides 2 are utilized. The *Channel Reduction* function is applied to subtract being *non-surface* logit from being *surface* logit for each voxel and we treat the difference as the logit of being *surface* within each column, and we select the voxel with the biggest being *surface* logit as the surface prediction.

The network is designed for the direct surface segmentation. The architecture consists of two main parts: a 3-D encoder-decoder *CNN* for surface probability map generation, and a trainable *CRFs* for modeling unary and pairwise simultaneously, as demonstrated in Fig. 6.

1) CNN: In medical image segmentation, the encoder-decoder *CNN* has been popular. We take a similar architecture to generate surface probability map, i.e. unary term. Global skip connections are built as in [2], as well as short or local skip connections utilized as in He *et al.* [17], where a unit block is called *residual* block. Those connections are used to mitigate gradient vanishing problem. As output of the encoder-decoder *CNN*, a two channel probability map for the target surface is generated. During pre-training the *CNN*, the supervision is added in with a *weighted binary cross-entropy* (WBCE) loss. The ground truth here is a binary mask of the same size as the input patch. The difference from those in *region* based segmentation neural network, is that 1 and 0 represent the target surface and background, respectively. Thus, the resulting classification problem is highly imbalanced. We introduce the WBCE loss to alleviate the problem.

2) CRFs: To explicitly model unary and pairwise simultaneously, the *CRFs* is introduced. To our best knowledge, commonly *CRFs* are used in *region* based segmentation network and it is the first time to apply it to the surface based segmentation. In Shah *et al.*'s method, a *FC* layer was utilized to directly regress the surface position but from feature maps in *low* spatial resolution.

TABLE I
LOSSES AND TRAINING STRATEGIES FOR THE PROPOSED SEGMENTATION METHODS.

	CNN Loss	CRFs Loss	Training Strategy
proposed CNN	WBCE	-	-
proposed CNN+CRFs		MCE	Pretrain CNN and then fine tune CNN+CRFs

The fully-connected *CRFs* model was first introduced to semantic segmentation by Krhenbhl and Koltun [15], which is known as *DenseCRF*. Although *DenseCRF* utilized a mean-field approximation inference, it achieved significantly improved results with an efficient inference. This has become the backbone for most *CRFs* models. The mean-field inference of a *DenseCRF* model can be incorporated into neural network, which was developed in [5]. This enables the joint training of *CNN* and *CRFs* by simple back propagation. This method was named as *CRF-as-RNN*. In *CRF-as-RNN*, the message-passing step is the bottleneck. The exact computation is quadratic in the number of pixels and therefore is infeasible. To alleviate this, a permutohedral lattice approximation was utilized. However, computing it efficiently on GPU is non-trivial or impossible to realize. In addition, an efficient gradient computation of the permutohedral lattice approximation, is also a non-trivial problem. This may hinder the learning of some parameters, e.g. θ_α, θ_β, and θ_γ. In the *convolutional CRFs* [18], the message passing is reformulated to be a convolution with a truncated Gaussian kernel and can be implemented in a similar way to the regular convolutions in CNN. Therefore the *convolutional CRFs* is utilized in the proposed method.

3) Loss: The *cross-entropy* (CE) loss is utilized both for the pre-training of the *CNN* and the fine tuning of the *CNN+CRFs* network. For pre-training, it is a *binary cross-entropy* (BCE) loss, since the encoder-decoder is meant to output probability map of being surface or non-surface. Also, as the surface pixel and non-surface pixel classes are highly imbalanced, a *weighted binary cross-entropy* (WBCE) loss is used. For the *CNN+CRFs* fine tuning, the problem is modeled as a multinomial classification and therefore a *multinomial cross-entropy* (MCE) loss is chosen. And the fine tuning is end-to-end. The losses and training strategies for the proposed method are summarized in Table. I.

In the following two sections, the proposed method was applied to the prostate MRI segmentation and the spleen CT segmentation.

III. APPLICATION TO THE PROSTATE MRI SEGMENTATION

A. Data, Patch Generation and Augmentation, Hyper Parameters

1) Data: The dataset is provided by the NCI-ISBI 2013 Challenge - Automated Segmentation of Prostate Structures [19]. This dataset has two labels: peripheral zone (PZ) and central gland (CG). We treat them both as prostate, since the single surface segmentation is considered in this work. The challenge data set has 3 parts including the training (60 cases), the leader board (10 cases) and the test data sets (10 cases). As the challenge is closed, only the training and leader board data with annotation, 70 cases in total, were used for our experiments. 10-fold cross validation was applied on this dataset.

2) Patch Generation: Our method needs pre-segmentation to set the desired topology and also to give the base plane for the 3-D patches, such that feature columns are sampled in normal directions. To test the robustness of the proposed method to pre-segmentation and the column length (the resolution is fixed), two pre-segmentation methods were explored. The first one was fitting a fixed size ellipsoid. The other one was coarsely fitting a mean shape to the user defined bounding box. Apparently the second one should produce more accurate pre-segmentations. And obviously we can sample shorter feature columns with a better pre-segmentation. All volumes were resampled to be isotropic with voxel resolution 0.625^3 mm^3 and normalized to have zero mean and unit variance.

a) Ellipsoid Pre-segmentation:: For simplicity, an ellipsoids, which has three principal semi-axes lengths as $25mm$, $22mm$ and $25mm$, was used as pre-segmentations. The centers of the ellipsoids were picked by users. As this pre-segmentation is far from perfect (average Dice similarity coefficient (DSC) around 0.7), longer columns should be sampled such that at least all surface voxels must be included. The column length for ellipsoid pre-segmentations was set to be 128 and the resolution was 0.625mm.

b) Mean Shape Pre-segmentation:: For training data, we aligned all images to one randomly picked reference image based on the ground truth centers, such that all training prostates were coarsely aligned. And then zero level set of average surface distance maps would be the mean shape. For test data, based on the bounding boxes users picked, we fitted the mean shape into the bounding boxes by only changing the value of level set, i.e. the mean shape was only allowed to do scaling transformation. The column length under this setting could be reduced to 64 (resolution=0.625mm) as they were more accurate (average DSC around 0.78).

3) Data Augmentation: Rotation (90°, 180°, 270°), flipping in two in-plane directions, combination of the two, and simple random translation in the z direction were applied. In total, the amount of the training patch was enlarged by a factor of 14, from 50×6 to $50 \times 6 \times 14$.

4) Hyper Parameters: The proposed network was implemented with Pytorch [20]. The network was initialized with Xavier normal initialization [21]. The patch size ($X \times Y \times Z$) was $32 \times 32 \times 128$ and $32 \times 32 \times 64$ with two different pre-segmentation settings, in which 32×32 represents the in-plane size or the number of columns in each patch, and the resolution on the column direction was 0.625mm.

a) The Proposed CNN only:: Adam optimizer [22] with learning rate 10^{-3}, was chosen for the training. We let it run for 50 epochs. The weights within the WBCE were $[1, 128]$ and $[1, 64]$ for patches of two different sizes, which are basically inversely proportional to the ratio between the non-surface voxels count and the surface voxels count in the ground truth annotation.

b) The Proposed CNN+CRFs:: The settings for the *CNN* pre-training were the same to that of *CNN* only. During fine tuning, the learning rate of Adam was 10^{-5}, and the training ran for 50 epochs. Only MCE loss following *CRFs* layer was utilized. The initialization of parameters in *CRFs* layer is detailed in Table. II.

TABLE II
INITIALIZATION OF PARAMETERS IN THE *CRFs* LAYER FOR THE PROSTATE MRI SEGMENTATION.

w_u	$w^{(1)}$	$w^{(2)}$	θ_α	θ_β	θ_γ	θ_{comp}
0.5	1	3	5	0.2	5	5

B. Evaluation Metrics

Three metrics – Dice similarity coefficient (DSC), Hausdorff distance (HD) (the greatest of all the distances from each point the computed surface to the closest point on the reference surface), and the average surface distance (ASD) (the average over the shortest distances between the points on computed surface and the reference surface), were engaged to evaluate results of segmentations. The definition of the DSC is:

$$DSC = \frac{2|V_g \cap V_p|}{|V_g| + |V_p|}, \tag{8}$$

where $|V_g|$ is the number of voxel of prostate in ground truth segmentation, $|V_p|$ is the prostate voxel number in prediction, and $|V_g \cap V_p|$ is the number of overlap prostate voxels between the ground truth and the prediction. The distance from a voxel s_1 to a surface S_2 is first defined as:

$$d(s_1, S_2) = \min_{s_2 \in S_2} ||s_1 - s_2||. \tag{9}$$

Then HD between the two surfaces S_1 and S_2 is computed as:

$$HD(S_1, S_2) = \max\{\max_{s_1 \in S_1} d(s_1, S_2), \max_{s_2 \in S_2} d(s_2, S_1)\}. \tag{10}$$

The ASD is defined as:

$$\begin{aligned} ASD(S_1, S_2) = \frac{1}{|S_1| + |S_2|}\big(&\sum_{s_1 \in S_1} d(s_1, S_2) \\ &+ \sum_{s_2 \in S_2} d(s_2, S_1)\big), \end{aligned} \tag{11}$$

where $|S_1|$ and $|S_2|$ are the number of voxels in surface S_1 and S_2, respectively.

C. Comparison to Other Methods

The quantitative segmentation results of different methods are listed in Table. III.

In Table. III, our results were derived using only NCI-ISBI data. While for compared methods: FCN [1], V-net [3], U-net [2], PSNet [23], they utilize additional in-house data and Promise12 data [24] for their network training. For all other methods, only NCI-ISBI dataset was used. In other words, the results of the first four methods were derived using around double training cases and a similar number of validation and test cases. With respect to the metric of DSC, our method outperforms FCN, V-net, U-net and PSNet, and are comparable to the deep learning state-of-the-art GCA-Net [25] and another state-of-the-art traditional method [26] combining the supervoxel method, Graph Cut and Active Contour Model (ACM). With respect to the surface distance related metrics (i.e., HD and ASD), the proposed method significantly outperform all the compared methods. Another advantage of the proposed method is that the post processing, e.g. morphological operations, to remove holes to which surface metrics are very sensitive, which is required for most *region* based deep learning methods such as FCN, V-net, U-net and so on, is not necessary any more. GS method shares the same merit. However, due to the need to manually design the cost function (how to generate the probability maps), even the solution to the energy minimization problem is global optimal, its performance is inferior to the proposed method and other deep learning based methods.

TABLE III
QUANTITATIVE COMPARISON WITH OTHER PROSTATE SEGMENTATION METHODS. THE FIRST FOUR METHODS USED BOTH THE NCI-ISBI DATASET AND THE PROMISE12 DATASET FOR TRAINING. THE REMAINING METHODS INCLUDING OUR METHOD ONLY USED THE NCI-ISBI DATASET.

	DSC	ASD (mm)	HD (mm)
FCN [1]	0.79±0.06	4.8±1.1	11.9±4.8
V-net [3]	0.83±0.05	3.4±1.2	9.5±3.9
U-net [2]	0.84±0.05	3.3±1.0	10.1±3.2
PSNet [23]	0.85±0.04	3.0±0.9	9.3±3.5
GS	0.80±0.04	2.7±0.6	13.9±1.8
SupervoxelGraphCutACM [26]	**0.88±0.02**	-	-
GCA-Net [25]	**0.88**	2.2	-
proposed CNN+CRFs	**0.88±0.03**	**1.4±0.3**	**8.2±3.6**

D. Robustness to Different Pre-segmentations

The results with two different pre-segmentations are shown in Table. IV. Better pre-segmentations and shorter columns improve the DSC and HD performance consistently. The ASD performances are comparable. The results basically indicate that although better pre-segmentations can help, our method is not sensitive but robust to different pre-segmentations if correct topologies are included.

TABLE IV
PROSTATE SEGMENTATION RESULTS OF THE PROPOSED METHODS WITH DIFFERENT PRE-SEGMENTATIONS AND COLUMN LENGTHS.

Pre-seg, Col length		DSC	ASD (mm)	HD (mm)
Ellipsoid, 128	proposed CNN+CRFs	0.86±0.05	**1.4±0.5**	9.6±5.2
Mean shape, 64	proposed CNN+CRFs	**0.88±0.03**	**1.4±0.3**	**8.2±3.6**

E. Ablation Study

We also investigated ablation study to verify the *CRFs* layer could improve the surface segmentation. The ablation study results are shown in Table. V. We compare results with or without *CRFs* layer. It could be noticed that the *CRFs* layer does not improve the DSC performance but it does help the surface related metrics performance. One sample of the improvement is illustrated in Fig. 7.

TABLE V
PROSTATE SEGMENTATION RESULTS WITH OR WITHOUT THE *CRFs*.

Pre-seg, Col length		DSC	ASD (mm)	HD (mm)
Ellipsoid, 128	proposed CNN	**0.86±0.05**	1.5±0.5	11.3±5.9
	proposed CNN+CRFs	**0.86±0.05**	**1.4±0.5**	**9.6±5.2**
Mean shape, 64	proposed CNN	**0.88±0.03**	**1.4±0.3**	8.3±2.9
	proposed CNN+CRFs	**0.88±0.03**	**1.4±0.3**	**8.2±3.6**

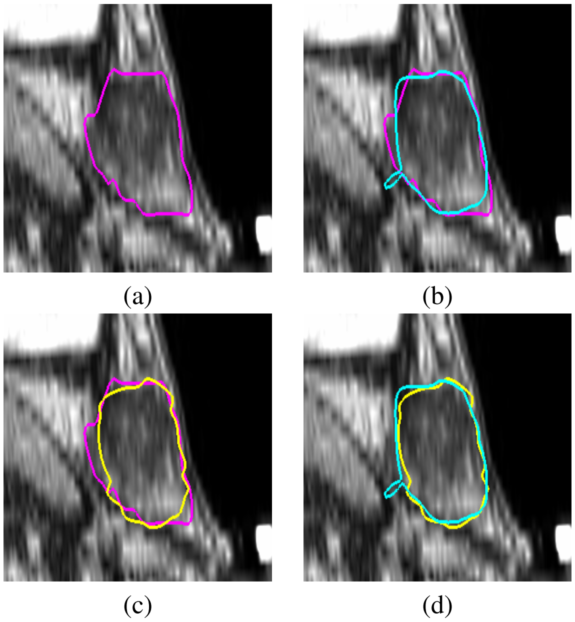

Fig. 7. Sample prostate segmentation results of proposed methods in the sagittal view. (a) The original image overlaid with the ground truth (purple). (b) The result of the proposed *CNN* (light blue). ASD and HD were 1.21mm and 13.8mm. (c)The result of the proposed *CNN+CRFs* (yellow). ASD and HD were improved to 1.20mm and 6.0mm. (d) Overlay of the result of *CNN* and that of the *CNN+CRFs*.

In the next section, the proposed method was tested in the spleen CT segmentation.

IV. APPLICATION TO THE SPLEEN CT SEGMENTATION

A. Data, Patch Generation and Augmentation, Hyper Parameters

1) Data: The dataset is provided by Task09 of Medical Segmentation Decathlon (MSD) challenge [1]. Only training sets with annotation were utilized. There are 41 cases in total. All experiments were conducted with 4-fold cross validation.

[1] https://decathlon.grand-challenge.org/Home/

2) Patch Generation: All volumes were resampled to be isotropic with voxel resolution 0.85^3 mm^3 and normalized to have zero mean and unit variance. For simplicity, a 3-D V-net was trained as the baseline model. A 3-D active contour model [27] was utilized to provide coarse segmentation. Then we smoothed the coarse predictions and treat the smoothed segmentations as our pre-segmentations. The generated patch has a size $32 \times 32 \times 128$.

3) Data Augmentation: The same augmentation strategy was applied to this task. In total, the training patch number is $26 \times 6 \times 14$.

4) Hyper Parameters: The proposed network was implemented with Pytorch. The network is initialized with Xavier normal initialization. The patch size ($X \times Y \times Z$) is $32 \times 32 \times 64$, in which 32×32 represents the in-plane size or the number of columns in each patch, and the resolution on the column direction is 0.85mm.

a) V-net: A public implementation [2] was used. The patch size is $128 \times 128 \times 64$. BCE is chosen as the loss function. The network was trained with Adam with learning rate 10^{-3} for 300 epochs.

b) The Proposed CNN+CRFs: For the *CNN* pre-training, Adam optimizer with learning rate 10^{-4}, was chosen. We let it run for 200 epochs. The weight within the WBCE is $[1, 64]$. During fine tuning, the MCE loss was utilized, and the learning rate of Adam is 10^{-5}, and the training runs for 100 epochs. The initialization of parameters in the *CRFs* layer is detailed in Table. VI.

TABLE VI
INITIALIZATION OF PARAMETERS IN *CRFs* LAYER FOR THE SPLEEN CT SEGMENTATION.

w_u	$w^{(1)}$	$w^{(2)}$	θ_α	θ_β	θ_γ	θ_{comp}
0.7	1	0.2	5	0.2	5	5

B. Evaluation Metrics

DSC, ASSD and HD were involved to quantify segmentation results.

C. Performance Comparison

The quantitative results of proposed method applied to the MSD Spleen dataset is shown in Table. VII. As the task is easy, it can be noticed that even the baseline V-net can achieve promising results. However, the proposed method can still gain additional improvement, especially in the sense of surface related metrics. From Table. VII, when considering p value of t-test, the proposed *CNN+CRFs* significantly outperforms V-net in aspects of DSC and ASD. Sample segmentation results are shown in Fig. 8. Illustrations that the *CRFs* improves segmentation results can be found in Fig. 9.

TABLE VII
QUANTITATIVE SEGMENTATION RESULTS COMPARISON FOR THE MSD SPLEEN DATASET.

	DSC	ASD (mm)	HD (mm)
V-net [3]	0.94±0.03	1.2±1.0	16.3±11.2
proposed CNN+CRFs	**0.95±0.02**	**0.86±0.74**	**13.6±12.7**
p-value	0.007	0.047	0.160

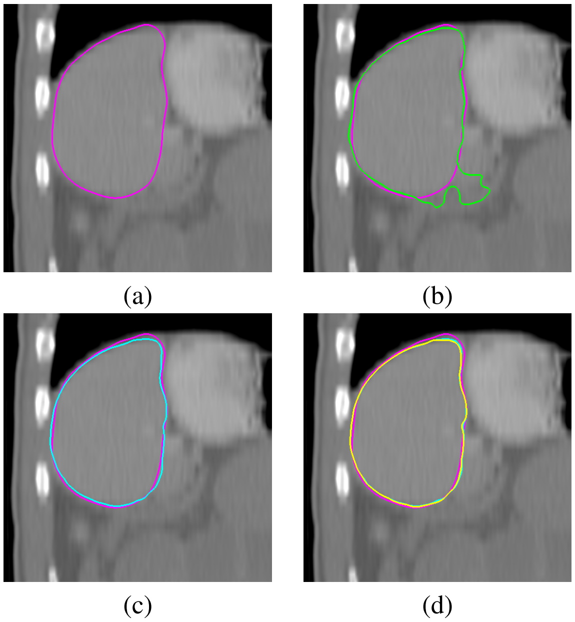

Fig. 8. Sample spleen segmentation results of proposed methods in sagittal view. (a) The original image overlaid with the ground truth (purple). (b) The result of V-net (geen). (c) The result of the proposed *CNN* (light blue). (d) The result of the proposed *CNN+CRFs* (yellow).

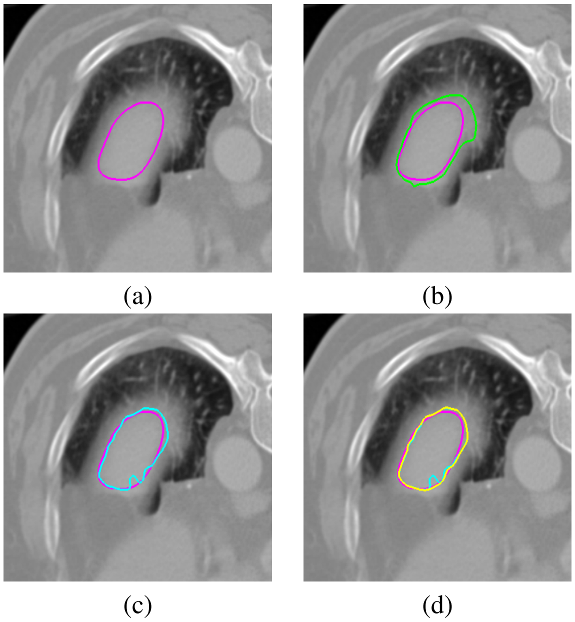

Fig. 9. Sample spleen slice where the *CRFs* helps. (a) The original image overlaid with the ground truth (purple). (b) The result of V-net (green). (c) The result of the proposed *CNN* (light blue). (d) The result of the proposed *CNN+CRFs* (yellow).

V. DISCUSSION

A. Training Efficiency

a) Patch Consistency: One advantage of proposed patch generation method is it generates more consistent patches, i.e.

[2] https://github.com/mattmacy/vnet.pytorch

all the patches' truths have monotonic surfaces within (each column has exact one voxel being surface). For the *region-based* network, the paradigms of truths for image patches generally have much more variance, e.g. all zeros (background), all ones (desired object) and any possibilities between. The proposed method may make the task easier for the network. Also the proposed method focuses on surfaces and segments surfaces directly, then the proposed network may predict boundaries/surfaces more accurately. One may think it is one kind of the attention mechanism [28].

b) Patch Amount: From the view of total patch number for each volume, compared to the common *region-based* 3-D segmentation network, the proposed method may be more efficient (actually we extract 6 patches for each volume), as we only sampled around pre-segmentation surface but not the whole volume. Surely it can be argued that if the pre-segmentation or ROI bounding box is given, the region or sub-volume based segmentation network may have similar number of patches.

Based on all these, if assuming similar augmentation utilized, the proposed method may converge faster. Actually the training of proposed *CNN* on the prostate data converges in around 3 hours with a Nvidia Titan X GPU.

B. Pre-segmenation with Correct Topology

The proposed method relies on the fact that the pre-segmentation must have a correct topology. Otherwise, there is no chance for our method to generate the correct prediction, since prediction of the proposed method always comply to the topology of the pre-segmentation. As was verified by experiments, the proposed method is not sensitive to the pre-segmentation accuracy as long as the topology is correct. In this sense, for application of simple topologies, model based pre-segmentation methods may work better than advanced methods without any topology guarantees. For example, a simple U-net/V-net may not be proper to use directly for the pre-segmentation generation, although the DSC of it's prediction may be significantly higher than that of the generated by a simple model based method. Actually, for the spleen dataset, we have to apply a recursive Gaussian mask smoothing and windowsinc mesh smoothing to get proper pre-segmentations.

C. Inference with Overlapping Patches

In our current implementation, each case only produces 6 patches, overlapping merely on the boundaries of patches. In *region-based CNN* segmentation work, it is commonly known that taking overlapping patches and averaging the prediction on the overlapped regions during inference can improve the segmentation results. Theoretically one can also take similar strategies in the *surface-based* segmentation and it probably helps to improve inference quality.

D. Possible Drawbacks of the Proposed Method

In the *region-based CNN+CRFs* framework, the visual feature is pixel or voxel intensity of original image, which is very helpful for the *CRFs* to recover the true object boundary accurately to compensate the coarseness character of *CNN-only* semantic segmentation. However, in our current *surface-based CNN+CRFs* framework, the original image information can not be used as in the *region-based* counterpart to help accurate boundary recovering. In GS framework, this problem is remedied by using carefully designed unary cost term, which includes enough lower level original image information. We may treat the problem as possible future work.

E. Extension to Surfaces with Complex Topologies

It is still open to apply the proposed method to application of single surface or object segmentation with complex topology, e.g. brain gray/white matter segmentation. For these applications, more carefully designed pre-segmentation method has to be considered. Also the sampling direction for image feature column may also need to be handled carefully such that no two columns have intersections. Possible options include the electric field line based method [29] and the generalized gradient vector flow based method [10].

F. Extension to Multiple Surfaces Segmentation

For multiple surfaces or objects segmentation, more efforts should be devoted on how to apply the proposed methods. 1) If all surfaces can share one pre-segmenation and image feature columns, it would be very straightforward to extend the proposed *CNN* from binary to multinomial. However, the problem becomes multi-label classification (each column has multiple labels, the number of which equals to the desired surface number), but not multi-class classification (each column has one label). The easier multi-class classification is properly modeled by current *CRFs* framework, which is not the case for the harder multi-label problem. 2) If surfaces can not share pre-segmentation and feature columns, how to reserve the topology conveyed by pre-segmentations would be non-trivial. For example, how to guarantee two surfaces not crossing each other, which is a very common prior knowledge in medical imaging. One may argue we can manage to control column length such that different surfaces can not intersect each other. But this moves the burden to decide proper column lengths. All these can be considered as the future work.

G. Loss Investigation

The MCE loss may not be the best option for our *surface-based* segmentation network, since the different possible labels for each column have ordering within. Therefore, in future we may consider weighted MCE, where the weights of different labels for each column should explicitly monotonically relates to the distances between current labels and ground truth label. Another possibility is to find out proper methods that can optimize surface position errors, e.g. mean square error, directly.

VI. Conclusion

We propose a novel direct surface segmentation method in 3-D using deep learning. With the proposed patch generation method, surface monotonicity with respect to the pre-segmentation is enforced in our segmentation neural network.

With an encoder-decoder network only, with respect to the surface distance related metrics (i.e., HD and ASD), the segmentation results on NCI-ISBI 2013 Prostate dataset and MSD Spleen dataset are promising. Together with the introduction of the *CRFs* layer into our deep network, the performance of the proposed surface segmentation method can even be improved further.

Acknowledgment

The authors would like to thank...

References

[1] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[2] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241.

[3] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *3D Vision (3DV), 2016 Fourth International Conference on*. IEEE, 2016, pp. 565–571.

[4] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Machine learning*, vol. 37, no. 2, pp. 183–233, 1999.

[5] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr, "Conditional random fields as recurrent neural networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1529–1537.

[6] X. Wu and D. Z. Chen, "Optimal net surface problems with applications," in *International Colloquium on Automata, Languages, and Programming*. Springer, 2002, pp. 1029–1042.

[7] 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, 2006.

[8] M. K. Garvin, M. D. Abramoff, X. Wu, S. R. Russell, T. L. Burns, and M. Sonka, "Automated 3-d intraretinal layer segmentation of macular spectral-domain optical coherence tomography images," *IEEE transactions on medical imaging*, vol. 28, no. 9, pp. 1436–1447, 2009.

[9] Y. Yin, X. Zhang, R. Williams, X. Wu, D. D. Anderson, and M. Sonka, "Logismoslayered optimal graph image segmentation of multiple objects and surfaces: cartilage segmentation in the knee joint," *IEEE transactions on medical imaging*, vol. 29, no. 12, pp. 2023–2037, 2010.

[10] I. Oguz and M. Sonka, "Logismos-b: layered optimal graph image segmentation of multiple objects and surfaces for the brain," *IEEE transactions on medical imaging*, vol. 33, no. 6, pp. 1220–1235, 2014.

[11] M. K. Garvin, M. D. Abràmoff, R. Kardon, S. R. Russell, X. Wu, and M. Sonka, "Intraretinal layer segmentation of macular optical coherence tomography images using optimal 3-d graph search," *IEEE transactions on medical imaging*, vol. 27, no. 10, pp. 1495–1505, 2008.

[12] Q. Song, J. Bai, M. K. Garvin, M. Sonka, J. M. Buatti, and X. Wu, "Optimal multiple surface segmentation with shape and context priors," *IEEE transactions on medical imaging*, vol. 32, no. 2, pp. 376–386, 2013.

[13] A. Shah, M. D. Abramoff, and X. Wu, "Simultaneous multiple surface segmentation using deep learning," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer, 2017, pp. 3–11.

[14] A. Shah, L. Zhou, M. D. Abrámoff, and X. Wu, "Multiple surface segmentation using convolution neural nets: application to retinal layer segmentation in oct images," *Biomedical optics express*, vol. 9, no. 9, pp. 4509–4526, 2018.

[15] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *Advances in neural information processing systems*, 2011, pp. 109–117.

[16] P. T. Choi, K. C. Lam, and L. M. Lui, "Flash: Fast landmark aligned spherical harmonic parameterization for genus-0 closed brain surfaces," *SIAM Journal on Imaging Sciences*, vol. 8, no. 1, pp. 67–94, 2015.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[18] M. T. Teichmann and R. Cipolla, "Convolutional crfs for semantic segmentation," *arXiv preprint arXiv:1805.04777*, 2018.

[19] N. Bloch, A. Madabhushi, H. Huisman *et al.*, "Nci-isbi 2013 challenge: automated segmentation of prostate structures," 2015.

[20] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *Advances in Neural Information Processing Systems*, 2017.

[21] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[23] Z. Tian, L. Liu, Z. Zhang, and B. Fei, "Psnet: prostate segmentation on mri based on a convolutional neural network," *Journal of Medical Imaging*, vol. 5, no. 2, p. 021208, 2018.

[24] G. Litjens, R. Toth, W. van de Ven, C. Hoeks, S. Kerkstra, B. van Ginneken, G. Vincent, G. Guillard, N. Birbeck, J. Zhang *et al.*, "Evaluation of prostate segmentation algorithms for mri: the promise12 challenge," *Medical image analysis*, vol. 18, no. 2, pp. 359–373, 2014.

[25] H. Jia, Y. Song, D. Zhang, H. Huang, D. Feng, M. Fulham, Y. Xia, and W. Cai, "3d global convolutional adversarial network for prostate mr volume segmentation," *arXiv preprint arXiv:1807.06742*, 2018.

[26] Z. Tian, L. Liu, Z. Zhang, J. Xue, and B. Fei, "A supervoxel-based segmentation method for prostate mr images," *Medical physics*, vol. 44, no. 2, pp. 558–569, 2017.

[27] Y. Gao, R. Kikinis, S. Bouix, M. Shenton, and A. Tannenbaum, "A 3d interactive multi-object segmentation tool using local robust statistics driven active contours," *Medical image analysis*, vol. 16, no. 6, pp. 1216–1227, 2012.

[28] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3640–3649.

[29] Y. Yin, Q. Song, and M. Sonka, "Electric field theory motivated graph construction for optimal medical image segmentation," in *International Workshop on Graph-Based Representations in Pattern Recognition*. Springer, 2009, pp. 334–342.