Simultaneous Detection of Multiple Elastic Surfaces with Application to Tumor Segmentation in CT Images

Kang Li[a], Marie-Pierre Jolly[b]

[a]Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213
[b]Siemens Corporate Research, Princeton, NJ 08540

ABSTRACT

We present a new semi-supervised method for segmenting multiple interrelated object boundaries with spherical topology in volumetric images. The core of our method is a novel graph-theoretic algorithm that simultaneously detects multiple surfaces under smoothness, distance, and elasticity constraints. The algorithm computes the global optimum of an objective function that incorporates boundary, regional and surface elasticity information. A single straight line drawn by the user in a cross-sectional slice is the sole user input, which roughly indicates the extent of the object. We employ a multi-seeded Dijkstra-based *range competition* algorithm to pre-segment the object on two orthogonal multiplanar reformatted (MPR) planes that pass through the input line. Based on the 2D pre-segmentation results, we estimate the object and background intensity histograms, and employ an adaptive mean-shift mode-seeking process on the object histogram to automatically determine the number of surface layers to be segmented. The final multiple-surface segmentation is performed in an ellipsoidal coordinate frame constructed by an automated ellipsoid fitting procedure. We apply our method to the segmentation of liver lesions with necrosis or calcification, and various other tumors in CT images. For liver tumor segmentation, our method can simultaneously delineate both tumor and necrosis boundaries. This capability is unprecedented and is valuable for cancer diagnosis, treatment planning, and evaluation.

Keywords: segmentation, multiple surfaces, surface-dependent cost, minimum *s*-*t* cut, range competition, tumor, necrosis, CT

1. INTRODUCTION

In the United States approximately 10,000 new patients are diagnosed with primary liver cancer (hepatocellular carcinoma) each year. The number of liver cancers diagnosed in the US and throughout the world is increasing at an alarming rate [1]. To diagnose, plan and evaluate cancer treatment, standard tumor size criteria (e.g., RECIST, WHO, or volume) need to be measured in medical images [2]. These measurements critically rely on an accurate segmentation of the tumor.

There has been relatively little prior work on liver lesion segmentation. The high variability in location, shape and appearance of liver tumors renders the segmentation task especially challenging. Previous approaches either work on two-dimensional images only [3], involve significant manual intervention [4], require a data-demanding training process [5], or rely on the integration of multiple imaging modalities (e.g., CT and PET) [6]. Most of these approaches focused on relatively simple hypodense liver tumors or metastases, which are nicely contrasted against the parenchyma [7]. Moreover, *none* of them considered the segmentation of necrotic or calcified tissue inside the tumor region, which provides valuable information to cancer treatment assessment [2].

We present a novel technique for segmenting multiple coupled object boundaries with spherical topology. The core component of our method is a graph-theoretic algorithm that simultaneously detects multiple coupled surfaces under geometrical constraints. The algorithm is a non-trivial extension of the framework proposed by Li *et al* [8, 9], Haeker *et al* [10, 11], and Wu *et al* [12]. In the previous framework, surface segmentation is formulated as a cost minimization problem with geometric constraints. The problem is modeled using a node-weighted directed graph. Each node of the graph encodes a unique spatial location in the image domain. The node weights define the cost function to be minimized, while the arc connections specify the constraints that

Contact information: `kangl@cmu.edu`, `marie-pierre.jolly@siemens.com`

limit the feasible configurations of the output surfaces. To enable an efficient solution, the arcs are constructed such that each closed set in the graph corresponds to a feasible surface (or a set of feasible surfaces); and the node weights are assigned such that the weight of each closed set (i.e., the total weight of nodes in the closed set) equals to the cost of the corresponding surface(s) (plus a constant). The closed set with the minimum weight, which correspond to the optimal surface(s), is then determined by computing a minimum s-t cut in an associated arc-weighted graph.

The cost function is a determining factor for the segmentation quality. As originally presented by Li *et al* [8,9] and in the follow up extensions by Haeker *et al* [10,11], the cost functions presented are all image-derived, and are independent of the surface configurations. However, in some applications, it may also be desirable to penalize for certain surface configurations or shapes, especially if certain prior knowledge about these properties is available. For example, we may prefer a flatter or rounder surface, if we knew that the object we are going to segment is flat or round. Or for the multi-surface case, we may want to penalize for the distances between the surfaces. This paper presents a way to incorporate such a surface-dependent cost, which we call the *elasticity* cost. To the best of our knowledge, this is the first time that a surface-dependent cost term has been incorporated into the graph-theoretic multi-surface segmentation framework. The term acts as an internal force on the surface, which penalizes deviation of the surface from an ellipsoid. We encode the term in a non-trivial fashion by adding a new set of weighted directed arcs to the graph. Analogous to the previous approach, the elastic surface detection problem can be transformed into a minimum *s-t cut* problem, which can be solved in polynomial time by various graph-theoretic algorithms [13] or via linear programming [14].

We integrated the graph-theoretic multi-surface segmentation algorithm into a semi-supervised tumor segmentation system for CT images. The system automatically detects the presence of necrosis and/or calcification in the tumor, and determines the number of surface layers to segment. The sole user input to the system is a single mouse drag over the tumor. We apply the system to the segmentation of liver lesions with necrosis or calcification, and various other tumors in CT images. We validate the method by comparing the segmentation results to expert-defined independent standard.

2. MULTIPLE ELASTIC SURFACE SEGMENTATION

2.1. Graph Modeling

Consider a volumetric image $I(x, y, z)$ of size $X \times Y \times Z^*$. We construct a graph $G = (N, A)$, which consists of a set of nodes N and a set of arcs A. We suppose that each surface to be segmented can be defined as a function $S(x, y)$ mapping (x, y) coordinates to z values. For simplicity, we also suppose that each node $n \in N$ in the graph uniquely corresponds to one voxel location in the image. Hence, any surface in I intersects with exactly one voxel (node) in each column of voxels (nodes) parallel to the z-axis, and it consists of exactly $X \times Y$ voxels (nodes). We denote the column of nodes at location (x, y) as $K(x, y)$. Each column has a set of neighbors. Typicall, we defined $K(x-1, y)$, $K(x+1, y)$, $K(x, y-1)$, and $K(x, y+1)$ as the neighbors of $K(x, y)$, but other neighborhood definitions are also possible. We refer to this model as a *multicolumn* graph model. To segment m coupled surfaces, the graph G consists of m disjoint subgraphs $\{G_i = (N_i, A_i) | i = 1, \ldots, m\}$. Each subgraph contains $X \times Y \times Z$ nodes, and is dedicated to segment one surface.

Note that the graph nodes and image voxels may not always be one-to-one correspondent. For instance, we may define the nodes to represent sub-voxel locations in order to achieve sub-voxel accuracy. In addition, to segment surfaces of other topology, such as spherical or cylindrical surfaces, we may either apply coordinate transformation to *unfold* the image [9], or construct the graph according to the object topology [8]. For example, to segment tumors, we employ elliptical coordinate transformation to unfold the input images.

We consider a surface *feasible* if it satisfies some application-specific *smoothness constraint*, defined by two parameters, Δ^x and Δ^y. Specifically, if (x, y, z) and $(x+1, y, z')$ are two neighboring points on a feasible surface, then $|z - z'| \leq \Delta^x$. Likewise, for (x, y, z) and $(x, y+1, z')$, we have $|z - z'| \leq \Delta^y$. If Δ^x (or Δ^y) is small, the feasible surfaces are smooth along the x (or y) direction, and the smoothness decreases with larger Δ^x (or Δ^y).

[*] We use $\mathbf{x}$ to denote the index set $\mathbf{x} \equiv \{0, \ldots, X-1\}$. Likewise for $\mathbf{y}$ and $\mathbf{z}$.

When segmenting m surfaces, we may enforce additional constraints to define the geometrical relations between the surfaces. For each pair of surfaces $S_i(x,y)$ and $S_j(x,y)$, where $1 \leq i,j \leq m$ and $i \neq j$, we require that $\delta^l_{ij} \leq S_i(x,y) - S_j(x,y) \leq \delta^u_{ij}$ for all (x,y). The parameters δ^l_{ij} and δ^u_{ij} ($\delta^u_{ij} \geq \delta^l_{ij}$) represent the surface *separation constraint*. The separation constraint define the range of signed distance between the surfaces. If δ^l_{ij} and δ^u_{ij} are both positive or both negative, the two surfaces do not intersect or overlap; otherwise, the surfaces are allowed to intersect and overlap. In this paper, we only consider the applications that require non-crossing surfaces. Therefore, we focus on the case where $\delta^u_{ij} > \delta^l_{ij} > 0$.

The above constraints are enforced in our graph model using arcs with infinite weights. Besides enforcing the constraints, the arc configuration ensures that each feasible surface uniquely corresponds to the upper envelop of a closed set in the graph. We construct three types of graph arcs, which include:

- Intra-column arcs A_a: The intra-column arcs connect the nodes in each column. Specifically, every node $n_i(x,y,z)$ $(z > 0)$ has a directed arc to the node $n_i(x,y,z-1)$, $i = 1,\ldots,m$, i.e.,

$$A_a = \{\langle n_i(x,y,z), n_i(x,y,z-1)\rangle \,|z > 0, \forall(x,y,i)\}, \tag{1}$$

  where $\langle n, n'\rangle$ denotes a directed arc from n to n'.

- Inter-column arcs A_r: The inter-column arcs encode the smoothness constraint, which is imposed between each pair of neighboring columns. Specifically, each node has directly arcs to the nodes with z-values that are Δ^x_i or Δ^y_i smaller in the neighboring columns:

$$\begin{aligned} A_r = &\{\langle n_i(x,y,z), n_i(x-1,y,\max(0,z-\Delta^x_i))\rangle \,|x \in [1, X-1], \forall(y,z,i)\} \\ &\cup\{\langle n_i(x,y,z), n_i(x+1,y,\max(0,z-\Delta^x_i))\rangle \,|x \in [0, X-2], \forall(y,z,i)\} \\ &\cup\{\langle n_i(x,y,z), n_i(x,y-1,\max(0,z-\Delta^y_i))\rangle \,|y \in [1, Y-1], \forall(x,z,i)\} \\ &\cup\{\langle n_i(x,y,z), n_i(x,y+1,\max(0,z-\Delta^y_i))\rangle \,|y \in [0, Y-2], \forall(x,z,i)\}. \end{aligned} \tag{2}$$

- Inter-surface arcs A_s: These arcs model the separation constraint between surfaces. For each pair of surfaces S_i and S_j, the inter-surfaces arcs connect the nodes in subgraphs G_i and those in G_j:

$$\begin{aligned} A_s = &\{\langle n_i(x,y,z), n_j(x,y,z-\delta^u_{ij})\rangle | \forall(x,y), z \geq \delta^u_{ij}, i \neq j\} \\ &\cup\{\langle n_j(x,y,z), n_i(x,y,z+\delta^l_{ij})\rangle | \forall(x,y), z < Z - \delta^l_{ij}, i \neq j\}. \end{aligned} \tag{3}$$

2.2. Incorporating Surface-Dependent (Elasticity) Costs

In the Section 2.1, we have discussed the approach to enforce hard geometrical constraints on the set of feasible surfaces. However, these constraints only confine the *range* of smoothness of each surface, and the *range* of separation distance between any two surfaces. They do not penalize for specific values of surface smoothness or separation. Hence, any sets of surfaces are equally feasible as long as their smoothness and separation parameters fall in the range of the constraints.

Now we go one step further. Suppose that in addition to the hard constraints, we also prefer surfaces that are smoother and closer together. Specifically, for each surface S_i, we define a smoothness cost function $p_i(|z - z'|)$ to penalize the difference in z-values between the surface points in different columns. Here, $z = S_i(x,y)$ and z' denote the z-value of the surface point on a neighboring column of $K(x,y)$. We also define a separation cost function $p_{ij}(|z_i - z_j|)$ for each pair of surfaces S_i and S_j, where $z_i = S_i(x,y)$ and $z_j = S_j(x,y)$. In general, we require the cost function $p_i(\cdot)$ to be convex and non-decreasing. A similar requirement is imposed on $p_{ij}(.)$. Without loss of generality, we assume that $p_i(0) = 0$ and $p_{ij}(\delta^l_{ij}) = 0$ (otherwise, we may simply subtract $p_i(0)$ or $p_{ij}(\delta^l_{ij})$ from the values of the cost functions, which does not affect their optima).

How do we incorporate the above surface-dependent cost functions into the graph model? Inspired by the work of Wu and Chen [12], we propose the following cost embedding scheme. In each subgraph $G_i \subset G$, we construct a directed arc from each node $n_i(x,y,z)$ to the nodes with z-values that are D^x_i or D^y_i smaller in the neighbor columns. Here $D_i^x = 0, \ldots, \Delta_i^x - 1$, and $D_i^y = 0, \ldots, \Delta_i^y - 1$. These arcs incorporate the smoothness cost functions. We call them the *smoothness-penalty* arcs A_{rp}:

$$\begin{aligned} A_{rp} =&\{\langle n_i(x,y,z), n_i(x-1,y,z-D_i^x)\rangle \,|1 \le x \le X-1, z \ge D_i^x, \forall(y,i)\} \\ &\cup\{\langle n_i(x,y,z), n_i(x+1,y,z-D_i^x)\rangle \,|0 \le x \le X-2, z \ge D_i^x, \forall(y,i)\} \\ &\cup\{\langle n_i(x,y,z), n_i(x,y-1,z-D_i^y)\rangle \,|1 \le y \le Y-1, z \ge D_i^y, \forall(x,i)\} \\ &\cup\{\langle n_i(x,y,z), n_i(x,y+1,z-D_i^y)\rangle \,|0 \le y \le Y-2, z \ge D_i^y, \forall(x,i)\}. \end{aligned} \tag{4}$$

Each arc $\langle n_i(x,y,z), n_i(x',y',z')\rangle$ in A_{rp} is assigned a finite weight $P_i(D)$, which only depends on the difference in z-values $D = |z - z'|$ between its incident nodes:

$$\begin{aligned} P_i(0) &= p_i(1) - p_i(0), \\ P_i(D+1) &= [p_i(D+2) - p_i(D+1)] - [p_i(D+1) - p_i(D)], \quad D \ge 0. \end{aligned} \tag{5}$$

Here, $P_i(D)$ is a discrete expression of the second derivative of $p_i(\cdot)$ at D, and is always non-negative.

To incorporate the separation costs, we construct a set of *separation-penalty* arcs A_{sp}. These arcs connect nodes $n_i(x,y,z)$ to nodes $n_j(x,y,z-d_{ij})$ with $\delta_{ij}^l \le d_{ij} < \delta_{ij}^u$, for each pair of surfaces S_i and S_j:

$$A_{sp} =\{\langle n_i(x,y,z), n_j(x,y,z-d_{ij})\rangle | \forall(x,y), z \ge d_{ij}, i \ne j\}. \tag{6}$$

Each separation-penalty arc $\langle n_i(x,y,z), n_j(x,y,z')\rangle$ is assigned a finite weight $P_{ij}(d)$, with $d = |z - z'|$:

$$\begin{aligned} P_{ij}(\delta_{ij}^l) &= p_{ij}(\delta_{ij}^l + 1) - p_{ij}(\delta_{ij}^l), \\ P_{ij}(d+1) &= [p_{ij}(d+2) - p_{ij}(d+1)] - [p_{ij}(d+1) - p_{ij}(d)], \quad \delta_{ij}^l \le d < \delta_{ij}^u. \end{aligned} \tag{7}$$

The rationale behind our arc construction will be elucidated in Section 2.4. The basic intuition is that a subset of the penalty arcs will be included in the minimum *s*-*t* cut, which we compute to obtain the optimal set of surfaces. Less smooth or more separate surfaces would cause more penalty arcs to be included in the cut set, and hence would be more expensive. The optimality of the surfaces depends on a balance between the surface-dependent costs and the image-derived costs (Section 2.3).

Note that, instead of favoring closer surfaces, we may also prefer surfaces that are more separated. In that case, the separation arcs connect nodes $n_i(x,y,z)$ to nodes $n_j(x,y,z+d_{ij})$ with $\delta_{ij}^l < d_{ij} \le \delta_{ij}^u$. The cost function $p_{ij}(.)$ is concave and non-increasing. The arc weights are assigned accordingly as:

$$\begin{aligned} P_{ij}(\delta_{ij}^u) &= p_{ij}(\delta_{ij}^u) - p_{ij}(\delta_{ij}^u + 1), \\ P_{ij}(d-1) &= [p_{ij}(d-1) - p_{ij}(d)] - [p_{ij}(d) - p_{ij}(d+1)], \quad \delta_{ij}^l < d \le \delta_{ij}^u. \end{aligned} \tag{8}$$

2.3. Incorporating Image-Derived Costs

We define the image-derived cost functions by assigning appropriate weights to the nodes of the graph. There are two types of image-derived costs: *on-surface* costs and *in-region* costs. For each voxel, the on-surface cost with respect to a target surface, denoted $C_{S_i}(x,y,z)$, indicates the unlikeliness of that voxel being on the surface; while the in-region cost with respect to a particular region, $C_{R_{i'}}(x,y,z)$, reflects the unlikeliness of that voxel belonging to the region. Since m surfaces $\{S_i | i = 1, \ldots, m\}$ divide an image domain into $m+1$ disjoint regions $\{R_{i'} | i' = 0, \ldots, m\}$, each voxel would be associated with $2m+1$ cost values. Here, regions R_0 and R_1 are separated by surface S_1, regions R_1 and R_2 are separated by S_2, and so on. The total cost of a set of feasible surfaces is then:

$$c = \sum_{i=1}^{m} \sum_{x,y} c_{S_i}(x,y,S_i(x,y)) + \sum_{i'=0}^{m} \sum_{x,y,z \in R_{i'}} c_{R_{i'}}(x,y,z) \tag{9}$$

We denote the weight of node $n(x, y, z)$ with respect to surface S_i as $w_i(x, y, z)$, where $i \in \{1, \ldots, m\}$. The weights are assigned such that the weight of every closed set in the graph equals to the cost of the corresponding set of surfaces (within a constant):

$$w_i(x, y, z) = w_{S_i}(x, y, z) + w_{R_i}(x, y, z), \tag{10}$$

where

$$w_{S_i}(x, y, z) = \begin{cases} c_{S_i}(x, y, z) & \text{if } z = 0, \\ c_{S_i}(x, y, z) - c_{S_i}(x, y, z-1) & \text{otherwise} \end{cases}, \tag{11}$$

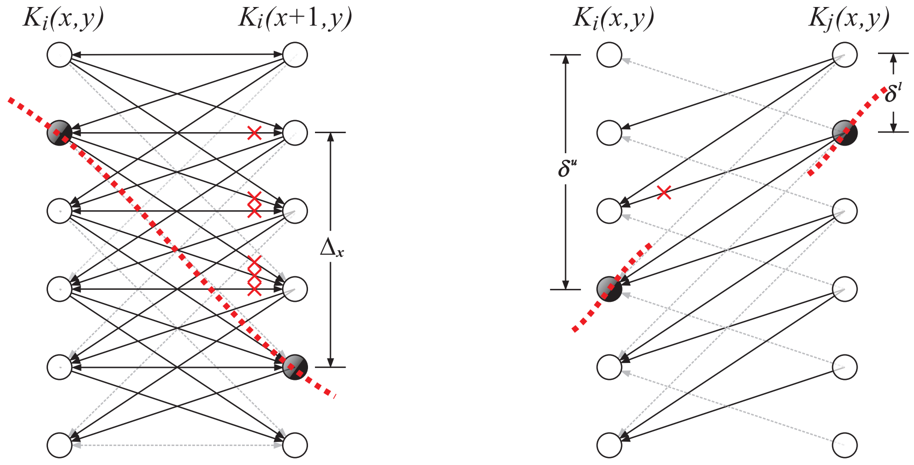
and

$$w_{R_i}(x, y, z) = c_{R_{i-1}}(x, y, z) - c_{R_i}(x, y, z). \tag{12}$$

We refer the reader to [10] for a more detailed discussion on the image-derived cost functions.

2.4. Computing the Optimal Surfaces

Once the graph is constructed and the costs assigned, we can use the same technique described in [9] to transform the graph into an arc-weighted s-t graph G_{st}. The transformation consists of three steps: 1) add two terminal nodes, a source node s and a sink node t, to the graph; 2) connect an arc weighted $-w$ from the source node to each non-terminal node with negative weight w; and 3) connect an arc weighted w from each non-terminal node with positive weight w to the sink. The node weights will be discarded, and the constraint arcs and penalty arcs are kept unchanged. We can then apply a minimum s-t cut algorithm to compute the optimal surfaces. The final surfaces will correspond to the upper envelope of the set of nodes that can be reached from s in G_{st}.



(a) Smoothness-penalty arcs

(b) Separation-penalty arcs

Figure 1. Illustration of penalty arcs. The gray dashed line represent constraint arcs, which have infinite weight. The solid lines represent penalty arcs. The arcs marked with red crosses will be contained in the s-t cut if the surface(s) pass through the black nodes.

During optimization, the penalty arcs effectively penalize for surfaces with less-prefered configurations by adding costs to the s-t cut. We illustrate the idea with two examples as shown in Figure 1. We suppose that (a) a feasible surface or (b) two feasible surfaces pass through the black nodes (marked by dotted lines). For each case, in order for the set of nodes under the surfaces to constitute a closed set, the arcs that are marked with red crosses must be cut off, and their weights will be added to the cost of the s-t cut. For the case of smoothness penalty, the total weight of the cut arcs is:

$$P_i(0) + [P_i(0) + P_i(1)] + [P_i(0) + P_i(1) + P_i(2)] = p_i(1) + [p_i(2) - p_i(1)] + [p_i(3) - p_i(2)] = p_i(3),$$

which equals to the penalty cost $p_i(3)$ for z-value difference 3 between the surface points. Likewise, in the separation penalty example, the weight of the cut arc equals to $p_{ij}(1) = 0$, which induces no penalty since the distance between the surface is already the minimum. Note that we have used the assumption that $p_i(0) = 0$ and $p_{ij}(\delta^l_{ij}) = 0$ (Section 2.2).

3. TUMOR SEGMENTATION

We integrated the graph-theoretic algorithm into a semi-supervised system for segmenting liver lesions with necrosis or calcification, and various other tumors in CT images. The system workflow is as follows.

1. The user initiates the segmentation by drawing a single line over the target object in a cross-sectional slice, which roughly indicates the extent of the object.

2. The system identifies another slice that passes through the input line and is perpendicular to the slice that the user drew upon. On each of the two slices, the system defines a circular region of interest (ROI) that encompasses the object in that slice.

3. Using the stroke and the ROI boundary as seeds, a multi-seeded range competition algorithm is used to pre-segment the ROI into object and background regions. The algorithm is loosely based on the multiseeded fuzzy connectedness algorithm proposed by Herman *et al* [15] and Udupa *et al* [16]. As a significant modification to the fuzzy connectedness method, the range of pixel intensities along a path is used as the path cost. As a result, the pixels whose intensities are within a shorter range to the object seeds (the stroke) than to the background seeds (the ROI boundary) will be segmented as the foreground object.

4. The object and background intensity histograms are estimated based on the 2D pre-segmentation results. We employ an adaptive mean-shift process [17,18] to identify the modes of the object histogram, and hence to determine the number of surface layers present in the target object.

5. The system constructs an ellipsoidal coordinate frame by fitting an ellipsoid to the pre-segmentation result. According to the number of surfaces determined in the previous step, the system executes a single- or double-surface search using the multi-surface segmentation algorithm, and outputs the final result. The final segmentation is performed in a narrow-band surrounding the fitted ellipsoid.

We now elaborate on the range-competition pre-segmentation and the final segmentation procedures.

3.1. Range Competition Pre-segmentation

The goal of this step is to produce an approximate segmentation of the lesion against the background in 2D planes in order to generate intensity histograms inside and outside the lesion. The input from the user is a mouse stroke through the lesion. Based on the stroke, the algorithm generates region seeds and background seeds. The region seeds are defined by the points on the stroke. The background seeds are placed along a circle with a diameter twice a large as the length of the stroke, as well as on the line segments that extend the stroke to the circle (see Figure 2). The range-competition algorithm computes the cost of the best path between a set of seeds and every pixel in the image. In a sense, it provides a measure of how strongly a pixel is connected to the seeds, or how much a pixel belongs to the region defined by the seeds.

Our approach is loosely based on the fuzzy connectedness algorithm [16]. We bring in two important modifications. One modification is in the definition of the cost function. In fuzzy connectedness, the cost of the path is traditionally defined as the maximum difference between neighboring pixels over the entire path. However, with this definition, it could happen that a path slowly deviates from the intensity of its origin while keeping a low cost, and eventually reaches a very different region. To prevent this, Udupa *et al* [16] proposed to constraint the pixel intensities along the path to follow a Gaussian distribution defined by the mean and variance of the intensities in the region of interest. For lesion segmentation, however, it is unrealistic to estimate these statistics ahead of time. Moreover, especially in the presence of necrosis and calcification, it is incorrect to assume that the intensity distributions are Gaussian.

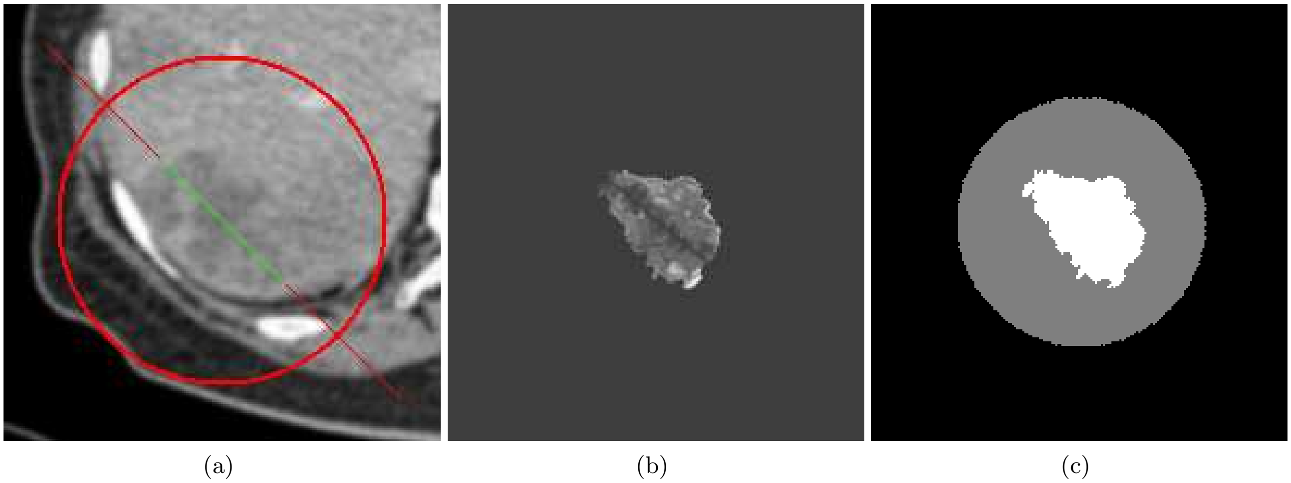


Figure 2. Multiseeded range competition. (a) Stroke seeds (green for region, red for background); (b) Lesion cost $C_L(x, y)$; (c) Region labels $L(x, y)$.

Instead, we define the cost function as the range of intensities along the path. The cost function is minimized using Dijkstra's algorithm. At each pixel (x, y), we keep track of the minimum intensity $F_{\min}(x, y)$, the maximum intensity $F_{\max}(x, y)$, and the cost $C(x, y)$ along the current best path to that pixel. At each iteration of Dijkstra's algorithm, the information at all neighbors (x', y') of (x, y) is updated by:

$$F_{\min}(x', y') = \min\left\{F_{\min}(x, y), I(x', y')\right\}, \quad F_{\max}(x', y') = \max\left\{F_{\max}(x, y), I(x', y')\right\},$$

$$\text{and } C(x', y') = F_{\max}(x', y') - F_{\min}(x', y'),$$

where $I(x', y')$ is the intensity at pixel (x', y'). This cost function favors paths that do not vary too much between the start pixel and the end pixel, and therefore stay within a homogeneous region. However, heterogeneous regions can still be recovered. Indeed, the stroke covers different parts of the lesion, and each point on the stroke can be responsible for gathering its own homogeneous part of the heterogeneous region.

The other modification is to use the multiseeded competition algorithm proposed by Herman *et al* [15]. This allows both foreground and background regions to compete as the paths to the pixels are being built. The algorithm outputs the cost image $C_L(x, y)$ which is the cost of reaching pixel (x, y) as well as a label image $L(x, y)$ such that $L(x, y) = 1$ if the pixel belongs to the lesion and $L(x, y) = 0$ otherwise. The lesion histogram is then built by combining the intensities of all the pixels labeled as part of the lesion.

3.2. Tumor Surface Layer Segmentation

To perform final segmentation, we first estimate the object and background intensity distributions based on the histograms obtained from the pre-segmentation. This is done by smoothing both histograms by a Parzen Window with a Gaussian kernel, and then normalizing them such that each of them sum to one. Depending on whether the estimated object intensity distribution is unimodal or not (as determined by the mean-shift process [17,18]), either a single-surface or a multiple-surface graph search will be executed. The graph search is performed in a narrow-band surrounding an ellipsoid fitted to the pre-segmentation result. The width of the narrow-band is set equal to one-fourth the length of the input stroke. We used the ellipsoidal coordinate transform to unfold the narrow-band to a Cartesian image $I(x, y, z)$, such that the tumor surfaces can be represented by a function $S(x, y)$. We incorporated on-surface, in-region, as well as surface-dependent cost functions as follow.

The on-surface costs for the outer (tumor) and inner (necrosis) layers are defined, respectively, as the negative gradient magnitude and the first derivative along the z-axis of the image:

$$C_{S_{\text{outer}}}(x,y,z) = -\sqrt{\left(\frac{\partial I(x,y,z)}{\partial x}\right)^2 + \left(\frac{\partial I(x,y,z)}{\partial y}\right)^2 + \left(\frac{\partial I(x,y,z)}{\partial z}\right)^2} \tag{13}$$

$$C_{S_{\text{inner}}}(x,y,z) = \frac{\partial I(x,y,z)}{\partial z} \tag{14}$$

If no necrosis is detected (i.e., the object intensity distribution is unimodal), only $C_{S_{\text{outer}}}$ is used.

The in-region costs are derived from the estimated intensity distributions. We denote the object and background intensity distributions by $\hat{H}_O$ and $\hat{H}_B$, respectively. If $\hat{H}_O$ is unimodal, we segment one surface only, and the cost function is given by:

$$C_R(x,y,z) = \hat{H}_B[I(x,y,z)] - \hat{H}_O[I(x,y,z)]. \tag{15}$$

Otherwise, we split $\hat{H}_O$ into two parts, $\hat{H}_{O_{\text{inner}}}$ and $\hat{H}_{O_{\text{outer}}}$, using Otsu thresholding [19]. The costs are then computed as:

$$C_{R_{\text{outer}}}(x,y,z) = \hat{H}_B[I(x,y,z)] - \hat{H}_{O_{\text{outer}}}[I(x,y,z)], \tag{16}$$

$$C_{R_{\text{inner}}}(x,y,z) = \hat{H}_{O_{\text{outer}}}[I(x,y,z)] - \hat{H}_{O_{\text{inner}}}[I(x,y,z)]. \tag{17}$$

We utilized the smoothness penalty cost only. It favors flatter surfaces, or equivalently, surfaces with less deviation from ellipsoids if transformed back to the original image coordinate. The cost function is defined as:

$$P_i(D) = D^2, \;\; i \in \{1,2\}, \;\; D = 0,1,2\ldots \tag{18}$$

4. EXPERIMENTS AND RESULTS

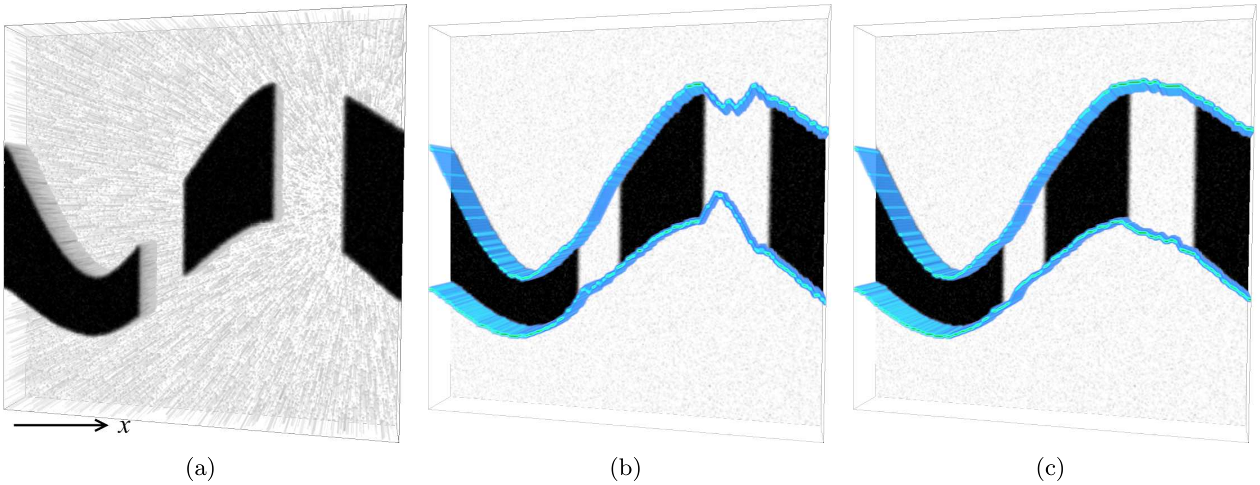


Figure 3. Double-surface segmentation results (blue ribbons) with and without incorporating the smoothness-penalty cost. (a) The volume rendering of the original image. (b) The segmentation result without incorporating smoothness penalty costs. (c) The segmentation result after incorporating the smoothness penalty.

We tested the proposed methods on both synthetic and CT images. To validate the correctness of the graphical model and to evaluate the effectiveness of the surface elasticity cost functions, we utilized a set of computer-synthesized images. Figure 3 shows one of the synthetic images, which illustrates the effect of the smoothness-penalty cost for segmenting the surfaces of a 3-D ridge with gaps. The image is of size $256 \times 256 \times 32$, and is superimposed with Gaussian noise (Figure 3(a)). We set the surface separation constraints as $\delta_l = 30$

and $\delta_u = 100$, which roughly equal to the surface distances at the thinnest and thickest parts of the ridge, respectively. The hard smoothness constraints were set to $\Delta_x = 2$ and $\Delta_y = 1$. The image-derived costs included the on-boundary costs only, which were computed as the negative gradient magnitudes. We first segmented the surfaces with only the image-derived costs. The segmentation result contains obvious fluctuations at the gaps (Figure 3(b)), where the optimal surfaces are vulnerable to noise. In fact, even if the image is noise-free, the optimal surfaces will still be biased toward the top edge of the image. This is because the value of the cost function vanishes at the gaps, and the solution of the optimal surface will favor locations with smaller z-values. (The origin of the coordinate system is at the top-left corner.) We then performed segmentation by incorporating the smoothness-penalty cost function given in Equation 18. The combination of this soft surface-dependent constraint and the hard geometrical constraints enabled us to obtain significantly improved results as shown in Figure 3(c), which clearly demonstrate the effectiveness of the cost function.

In addition to synthetic images, we collected 15 CT datasets from patients with various types of tumors. The datasets were acquired at 9 different clinical sites with different imaging protocols, and the slice thicknesses range from 1 to 5 mm. Expert radiologists at the different sites identified and manually segmented a total of 57 lesions in these datasets, with and without necrosis/calcification. Among these lesions, there are 15 liver lesions, and 42 other lesions in lungs, pancreas, colon, ovaries, rectum, etc. The lesions ranged in size from 0.1 to 700 ml. Examples of the segmentation results on cross-sectional planes are shown in Figure 4.

To evaluate the performance of the method for tumor and necrosis surfaces segmentation, we computed a set of quantitative indices using the manually-defined ground truth. These indices include the mean and maximum unsigned border positioning errors, as well as the area overlap ratio. The area overlap ratio is computed as $\alpha = (A_{\mathrm{A}} \cap A_{\mathrm{GT}})/(A_{\mathrm{A}} \cup A_{\mathrm{GT}})$, where A_{A} is the automatically segmented area, and A_{GT} is the ground-truth area. All indices are reported as mean ± standard deviation in pixels. The validation was performed on 2-D cross-sectional planes only. The results are summarized in Table 1. In total, our system correctly identified the (non)existence of necrotic and/or calcified tissues in 49 tumors. It failed to identify necrosis in 5 tumors, and falsely detected necrosis in 3 tumors that do not contain necrotic tissue.

Table 1. Summary of Segmentation Performance Indices for 57 Tumors

Surface	Mean Unsigned Error	Maximum Unsigned Error	Area Overlap
Lesion	1.36±1.04	5.02±2.75	87.19%±5.53%
Necrosis/Calcification	1.42±0.69	5.77±2.45	80.59%±9.59%

5. CONCLUSION

We presented a novel approach to incorporate surface elasticity into the objective function of a graph-theoretic approach for segmentation of multiple closed surfaces in volumetric images. This information is encoded in a non-trivial fashion by adding a new set of weighted directed arcs to the graph. The combination of one-stroke user interface, range competition pre-segmentation, mean-shift mode seeking, and segmentation in ellipsoidal coordinates is original. The capability to simultaneously identify liver tumor and necrosis boundaries is unprecedented. We show that the incorporation of surface elasticity term into the optimal multiple coupled surface segmentation algorithm, when combined with the previously-published on-boundary and in-region terms, improves the segmentation result. We demonstrate the power and ease of use of our method for tumor segmentation in CT image, and quantitatively validate the segmentation results. The results obtained by our method build an important basis for quantitative measurement of tumor characteristics, including RECIST, WHO, necrosis, and volume measurements, which are valuable for cancer diagnosis, treatment planning, and evaluation.

REFERENCES

1. "The Johns Hopkins liver cancer web page," 2006.
2. F. H. Miller, A. L. Keppke, D. Reddy, J. Huang, J. Jin, M. F. Mulcahy, and R. Salem, "Response of liver metastases after treatment with yttrium-90 microspheres: Role of size, necrosis, and pet," *American Journal of Roentgenology* **188**, pp. 776–783, June 2007.
3. C. Krishnamurthy, J. J. Rodriguez, and R. J. Gillies, "Snake-based liver lesion segmentation," in *Proc. 6th IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 187–191, Mar. 2004.
4. R. Lu, P. Marziliano, and C. H. Thng, "Liver tumor volume estimation by semi-automatic segmentation method," in *Proc. 27th Annual International Conference of the Engineering in Medicine and Biology Society (IEEE-EMBS)*, pp. 3296–3299, Sept. 2005.
5. Y. Li, S. Hara, and K. Shimura, "A machine learning approach for locating boundaries of liver tumors in ct images," in *Proc. International Conference on Pattern Recognition (ICPR)*, pp. 400–403, Aug. 2006.
6. V. Potesil, X. Huang, and X. S. Zhou, "Automated tumor delineation using joint PET/CT information," in *Proc. SPIE International Symposium on Medical Imaging: Computer-Aided Diagnosis*, **6514**, Mar. 2007.
7. M. Bilello, S. B. Gokturk, T. Desser, S. Napel, R. B. Jeffrey Jr., and C. F. Beaulieu, "Automatic detection and classification of hypdense hepatic lesions on contrast-enhanced venous-phase CT," *Medical Physics* **31**(9), pp. 2584–2593, 2004.
8. K. Li, S. Millington, X. Wu, D. Z. Chen, and M. Sonka., "Simultaneous segmentation of multiple closed surfaces using optimal graph searching," in *Proc. International Conference on Information Processing in Medical Imaging (IPMI)*, pp. 406–417, Springer, July 2005.
9. K. Li, X. Wu, D. Z. Chen, and M. Sonka, "Optimal surface segmentation in volumetric images – a graph-theoretic approach," *IEEE Trans. Pattern Anal. Machine Intell.* **28**, pp. 119–134, Jan. 2006.
10. M. Haeker, X. Wu, M. D. Abrámoff, R. Kardon, and M. Sonka, "Incorporation of regional information in optimal 3-d graph search with application for intraretinal layer segmentation of optical coherence tomography images," in *Proc. International Conference on Information Processing in Medical Imaging (IPMI)*, pp. 607–618, Springer, July 2007.
11. M. Haeker, M. D. Abrámoff, X. Wu, R. Kardon, and M. Sonka, "Use of varying constraints in optimal 3-D graph search for segmentation of macular optical coherence tomography images," in *Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention*, **10**, pp. 244–251, Springer, Oct. 2007.
12. X. Wu and D. Z. Chen, "Optimal net surface problems with applications," in *Proc. 29th International Colloquium on Automata, Languages and Programming (ICALP)*, pp. 1029–1042, July 2002.
13. Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Machine Intell.* **26**, pp. 1124–1137, Sept. 2004.
14. R. D. Carr, G. Konjevod, G. Little, V. Natarajan, and O. Parekh, "Compacting cuts: A new linear formulation for minimum cut," in *Proc. 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 43–52, Jan. 2007.
15. G. T. Herman and B. M. Carvalho, "Multiseeded segmentation using fuzzy connectedness," *IEEE Trans. Pattern Anal. Machine Intell.* **23**, pp. 460–474, May 2001.
16. J. K. Udupa and S. Samarasekera, "Fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation," *Graphical Models and Image Processing* **58**(3), pp. 246–261, 1996.
17. Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Trans. Pattern Anal. Machine Intell.* **17**, pp. 790–799, Aug. 1995.
18. M. Singh and N. Ahuja, "Regression based bandwidth selection for segmentation using parzen windows," in *Proc. IEEE International Conference on Computer Vision*, **1**, pp. 2–9, Oct. 2003.
19. N. Otsu, "A threshold selection method from gray level histograms," *IEEE Trans. Syst., Man, Cybern.* **9**, pp. 62–66, Mar. 1979.

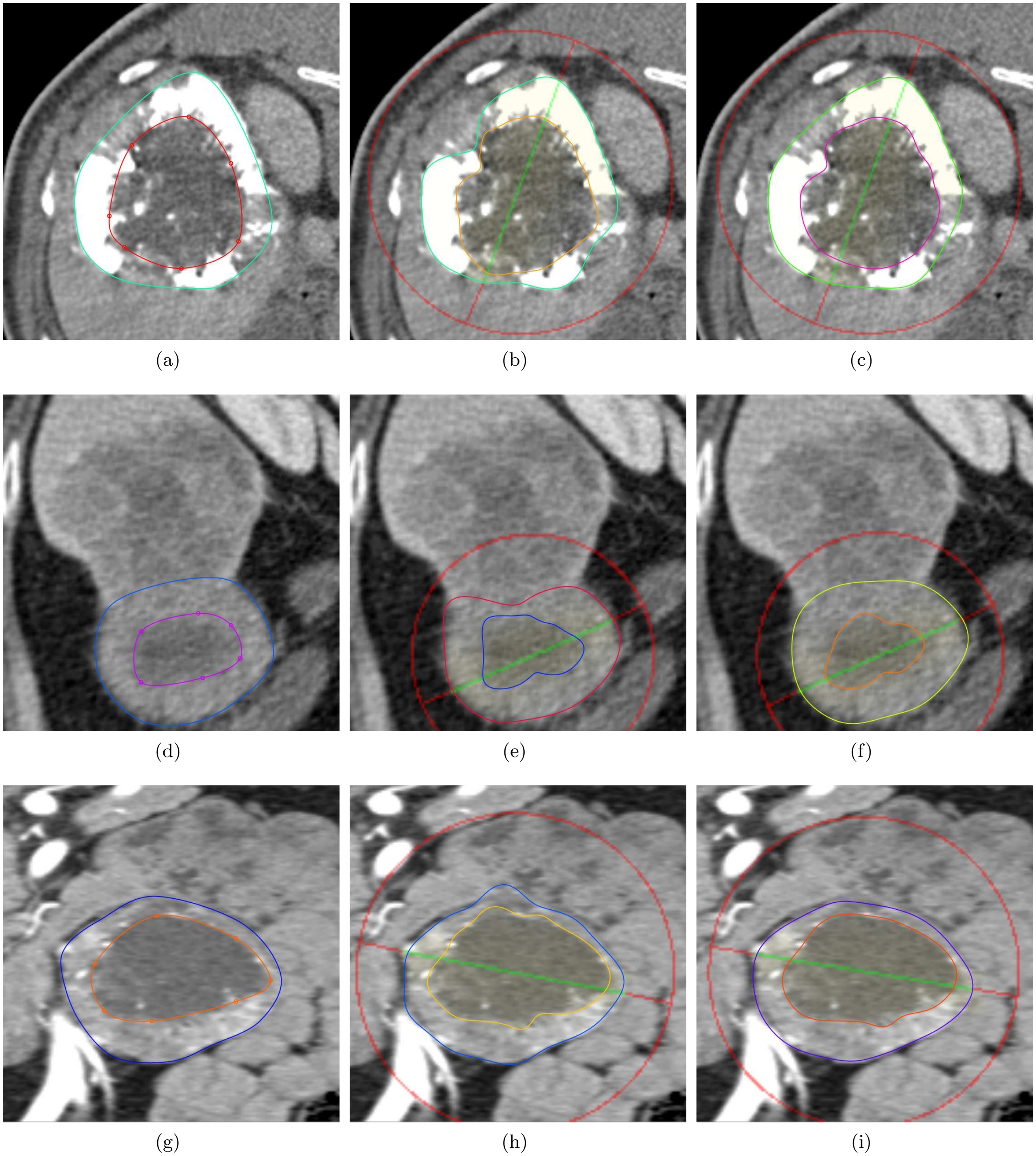


Figure 4. Examples of tumor segmentation results on cross-sectional planes. Green straight line segments indicate the user-input strokes. Red circles are the automatically-determined region of interest. (a),(b),(c) A liver lesion with surrounding calcification. (d),(e),(f) A liver lesion with a necrotic core. (g),(h),(i) A paraganglioma. The first column shows the original images; the second column shows the segmentation without incorporating elasticity costs; and the third column shows the segmentation with smoothness penalty costs incorporated.