

# A SPLICE-GUIDED DATA DRIVEN INTERACTIVE EDITING

Noha El-Zehiry Marie-Pierre Jolly Michal Sofka

Imaging and Computer Vision  
Siemens Corporation, Corporate Technology, Princeton, USA

## ABSTRACT

Image segmentation is one of the most challenging tasks in the field of image processing. Even the best automatic segmentation approaches cannot yet provide accurate segmentation in all situations. Hence, there is a persistent need for interactive editing tools to correct the automatic segmentation results such that they match what would be clinically accepted by an expert. We present an editing approach that uses a user-drawn splice (contour) in 2D to correct any 2D or 3D segmentation that may have been obtained automatically or manually. The algorithm integrates the image data, the existing segmentation (presegmentation), and the user's input into an energy minimization framework. We will show that the proposed segmentation editing approach is general and can be used in multiple applications and for multiple imaging modalities.

## 1. INTRODUCTION

Interactive editing tools are necessary for many medical image analysis systems that generate segmentations by an automatic algorithm or with the help of a user. These tools are used for refining the initial segmentations with the following requirements: (1) follow the boundaries of anatomical structures more accurately, (2) capture the level of detail required by a particular application, (3) correct errors in the initial segmentation results, and (4) include or exclude parts of anatomical structures. The final edited segmentation is approved by a clinician and used for subsequent diagnosis and treatment planning.

Interactive editing approaches have gained more attention recently with the growing number of applications where segmentation refinement is necessary. Grady and Funka-Lea [3] cast the editing approach as an energy minimization problem that integrates the image data, the original segmentation, and the user input seeds (pixels/voxels marked as foreground or background). The seeds are added inside and/or outside the object of interest which is not practical, since clinicians are typically trained to highlight object boundaries. Heckel *et al.* [4] presented an approach that generates 3D update using live-wire extrapolation of user drawn boundary contour in 2D. However, the algorithm does not make use of presegmentation and it is computationally expensive (an update on a small object, such as a lesion, take 44 second). Algorithms that ignore presegmentation (eg. [4], [8]) do not perform editing that benefits from the current segmentation but rather it solves a new segmentation problem in a neighborhood defined by the user input. In [4],

the segmentation problem is solved using a live-wire framework [1] and in [8], it is solved using graph cuts [2].

In this paper, we present a segmentation editing approach based on user-drawn free form contour, *splice*, which highlights part of the object boundary. The editing tool integrates all possible information (data, presegmentation, and user input) to produce fast 3D updates. The updates are efficiently and effectively computed not only for small anatomical structures (e.g. prostate or tumors) but also for large structures (e.g. liver). We will show several such editing cases on challenging MR and CT volumes. The algorithm reduces the average point-to-surface segmentation error by 15% and reduces the average error by 27% compared to the most recent automatic segmentation approaches [7].

## 2. METHODS

We start by presenting the formulation and then we proceed by giving a detailed description for the *splice editing* algorithm. The splice editing algorithm combines the existing segmentation, the original data, and the input from the user. In case any of these inputs is missing, the algorithm still provides useful update as will be apparent from the formulation. The algorithm uses a simple input, a 2D splice and generates 3D updates of the segmented volume. In this paper, the splice editing is formulated as an energy minimization problem where a graph based approach is used for the optimization of this energy. A Graph  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$  consists of a set of vertices  $v \in \mathcal{V}$  and a set of edges  $e \in \mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ . An edge incident to vertices  $v_i$  and  $v_j$  is denoted  $e_{ij}$ . A weighted graph is a graph in which every edge  $e_{ij}$  is assigned a weight  $w_{ij}$ . An edge cut is the set of edges whose removal disconnects the graph. We employ the edge weights to reflect the energy terms. Each voxel  $i = (x, y, z)$  is associated with a binary variable  $x_i$  that represents the label of  $i$  ( $x_i = 1$  if  $i \in \omega$  (the object of interest), and 0 otherwise). Every variable  $x_i$  has a corresponding vertex  $v_i$  in the graph  $\mathcal{G}$  and each edge is used to connect vertices in a local neighborhood. Although the experiments in this paper were performed using a 6-connected Cartesian lattice, the formulation in section 2.1 is general and can be applied to any graph.

### 2.1. Data Driven Editing

The information from the presegmentation, the user's input and the image intensity is integrated to provide an update (edit) of the available segmentation. We first review the editing model [3] and then we describe how to generate seeds from the user

splice. Let's define a presegmentation,  $p$ , determined by another process (e.g., an automatic segmentation algorithm), as

$$p_i = \begin{cases} 1 & \text{if } v_i \text{ was presegmented as foreground,} \\ 0 & \text{if } v_i \text{ was presegmented as background.} \end{cases} \quad (1)$$

The editing problem may be defined as the minimization of the energy functional

$$Q(x) = \sum_{e_{ij}} w_{ij} (x_i - x_j)^2 + \gamma_i \left( \sum_i (1-p_i) x_i + \sum_i p_i (1-x_i) \right), \quad (2)$$

with respect to the foreground indicator function  $x$ , defined on the vertices of the graph  $\mathcal{G}$ , where  $\gamma_i$  is a parameter indicating the strength of the presegmentation,  $\gamma$  depends on the distance of a particular voxel  $v_i$  from the foreground seeds  $F$  and the background seeds  $B$  and can be defined as:

$$\gamma_i = \kappa \exp \left( -\frac{d(v_i, v_j)}{\sigma} \right), \quad (3)$$

where  $d(v_i, v_j)$  is the minimum distance from  $v_i$  to all the seed vertices  $v_j \in F, B$ . Hence,  $\kappa$  is a parameter that indicates the overall strength, and  $\sigma$  reflects the domain of influence of the seeds. In this paper we calculate  $\sigma$  as:

$$\sigma = \frac{1}{2} \left( \frac{3 \text{Vol}}{4\pi} \right)^{\frac{1}{3}}. \quad (4)$$

where Vol is the volume of the object of interest to be edited. The weights  $w_{ij}$  represent the contrast weighting of the graph based on the intensity values. The weights  $w_{ij}$  are defined as

$$w_{ij} = \exp(-\beta(g_i - g_j)^2), \quad (5)$$

where  $g_i$  represents the gray scale intensity at vertex (voxel)  $v_i$  and  $\beta$  is a free parameter that controls the contribution of the contrast information.

This energy has two competing terms: data-driven smoothness in the form of the first term and presegmentation in the form of the second term. The user input is included through the set of vertices (possibly empty) marked as foreground seeds  $F \subset V$  and a set of vertices (possibly empty) marked as background seeds,  $B \subset V$ , such that  $F \cap B = \emptyset$ . The seeds are incorporated into the minimization of (2) by performing a constrained minimization of  $Q(x)$  with respect to the constraints  $x_i = 1$  if  $v_i \in F$  and  $x_i = 0$  if  $v_i \in B$ . In our paper, we devise a system to generate these inputs (seeds) based on a splice interaction favored by clinicians over seed interaction.

A major advantage of our algorithm over the data driven editing presented in [3] is the splice interaction that we are presenting in this paper. Clinicians favor to draw splices close to the boundaries of the object because they are trained to visually find such boundaries when reading scans. It is less confusing to draw splices than placing brush seeds inside and outside the object of interest as in [3]. However, since the sets  $F$  and  $B$  are not given explicitly by the user, this poses a question how to generate them reliably. The system we design in this paper generates the seeds based on the input splice (Section 2.2).

The splice-driven editing has several advantages with the direct impact on the quality and the speed of the editing. The

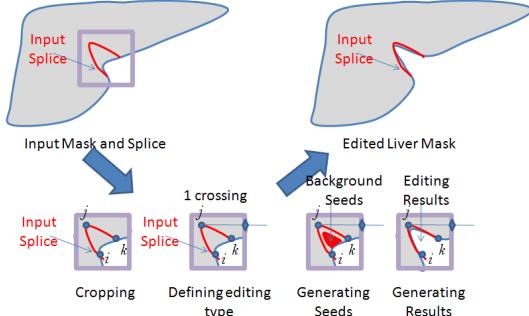
segmentation results are updated quickly – once the user releases the mouse. This is because the editing is confined to a small region of interest surrounding the splice and therefore the optimization in (2) is *local*. In contrast, manually placed seeds may be spread across the volume which results in a large region of interest. To maintain low computational cost when manually placed seeds are used [3], the image and presegmentation regions have to be downsampled, which has two effects. First, optimization in (2) is performed on the *entire* region which may introduce undesired edits away from the user's input. Second, the details in the editing update are lost due to coarse resolution of the input. The locality of our approach overcomes the aforementioned problems. Since our editing is local, the radius  $\sigma$  in (4) of the editing influence is defined to be object dependent. This is a key contribution to make the algorithm work well for small objects (e.g. tumors) and large objects (e.g. liver) without additional parameter tuning. Nevertheless, if the splice is really large yielding a cropped volume that exceeds  $10^6$  voxels, we apply downsampling.

## 2.2. Splice Editing Algorithm

One of the greatest challenges when using the splice interface in conjunction with the energy model in (2) is the definition of the foreground and/or background seeds. This section provides description of the seed generation and the editing. The step-by-step splice editing algorithms (illustrated in Figure 1) is described as follows:

1. Acquire the input splice (a set of points drawn in a 2D view by the user).
2. Calculate the extent of the input splice (i.e. the maximum and minimum values for its x,y and z coordinates).
3. Crop around the extent of the splice after adding a safety margin in each direction.
4. Downsample the cropped volume, if necessary.
5. Find the intersections ( $i$  and  $k$ ) of the splice with the projection of the presegmentation mesh (the boundary of the input mask) in the slice where the splice was drawn.
6. Find the farthest point,  $j$ , (on the splice) from the presegmentation mesh.
7. Apply a ray crossing test [5] originating at point  $k$  to test if  $k$  falls inside or outside the presegmentation mask.
8. The seeds are generated as follows:
  - (a) For each voxel  $v$  in the cropped box, calculate the minimum distance  $d(v)$  to the splice.
  - (b) If  $j$  is inside the presegmentation mask and  $d(v) > th^1$  add a background seed to remove the undesired parts from the presegmentation mask.
  - (c) If  $j$  is outside the presegmentation mask and  $d(v) > th$  add a foreground seed to add the desired voxels to the presegmentation mask.
9. Use the generated seeds and the constructed graph with the weighting in Section 2.1 and run the random walker algorithm as in [3] to generate an updated segmentation.

<sup>1</sup>The threshold  $th$  is determined experimentally (we used a threshold of 2 voxels). Notice that if the threshold is set to zero then the update is forced to pass through the splice that the user has drawn.



**Fig. 1.** Steps of Splice Editing

10. For every subsequent splice, apply steps 1-9 on the updated segmentation mask.

### 3. RESULTS

The gallery of results in Figure 2 illustrates that our editing works well for different applications and can be applied to different modalities. It also shows that the algorithm maintains its effectiveness even when the data discrimination vanishes.

Figure 2 (a) shows an example of editing the liver. It depicts an abdominal MR scan of the liver and the segmented liver using an automatic segmentation method from [7]. The detected liver mesh deviates from the correct liver boundary due to a prior shape constraint that was enforced in the automatic segmentation. The second image shows the input splice drawn in yellow and the updated mesh that coincides with the correct liver boundary. The example in Figure 2 (b) shows an abdominal CT scan and the detected liver mesh. The vena cava is falsely labeled as part of the liver by the automatic segmentation algorithm. Our interactive editing scheme can correct the automatic segmentation and the vena cava can be excluded from the liver mesh by a single splice as depicted in the second image. Figure 2 (b) shows 5 subsequent slices<sup>2</sup> of the edited mesh: the slice where the splice was drawn, the two preceding and the two following slices to the spliced slice. These images demonstrate that the update is performed in 3D, although the input is a single 2D splice.

The examples illustrate the strength of having both presegmentation and data terms in the formulation in (2). In the MR example, there is a strong contrast and hence incorporating the data information is useful and plays a major role in producing the correct editing. On the other hand, in the CT example, there is almost no contrast between the vena cava and the tissue liver and hence, the algorithm depends mainly on the user's input and the presegmentation to provide the correct editing result.

Figure 2 (c) shows prostate segmentation in a CT scan. The editing in this example is performed using two splice updates. The first splice at the top of the segmented mesh is used to add more voxels to the foreground area (the prostate) and the second splice drawn is used to remove the oversegmented part at the bottom of the prostate.

Splice-guided editing can also be used as a virtual scalpel to remove parts of segmentations for resection surgery plan-

ning. The user draws a splice around the tumor that should be resected and the editing tool provides a 3D resection surface that excludes the tumor from the final segmentation. Two such examples are illustrated in Figure 2 (d). The first example shows a wedge resection for a liver tumor and the second example shows the resection of lung tumors. The splices are drawn slightly away from the tumor boundaries to account for a resection safety margin around tumors.

To evaluate the robustness of our algorithm, we ran a quantitative evaluation on a 30 data sets of MR liver images. The automatic segmentations are calculated using the approach in [7]. We compared the segmentation accuracy of the edited volumes to the automatically segmented volumes. Our editing algorithm improves the average segmentation error by 15 % and the maximum segmentation error by 27 %. The error measure that we used is the mean point-to-surface error using 2000 points. The average number of splices required to correct each data set is 16 splice and every update is performed in 2 seconds.

### 4. CONCLUSION AND FUTURE WORK

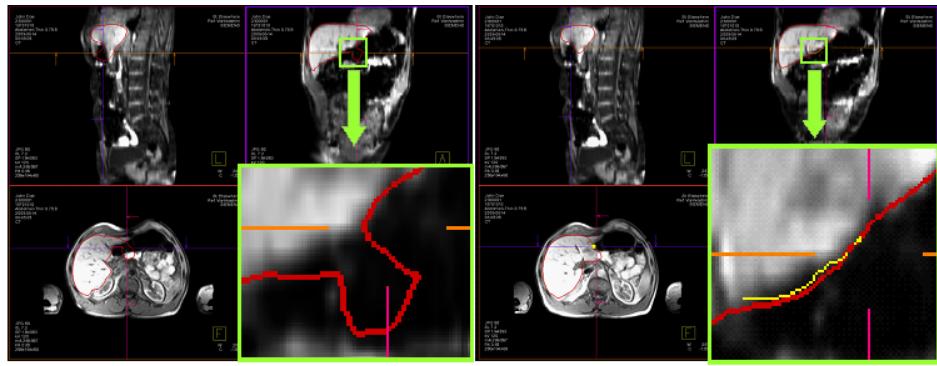
The paper presented an interactive editing tool that uses a splice interface which is intuitive and easy to use by clinicians. We use an energy minimization framework to integrate presegmentation, data, and user interaction to produce final 3D update. The qualitative and quantitative evaluation showed that the average segmentation error is improved by 15% and the maximum segmentation error by 27% compared to the state-of-the-art segmentation approach. In the future, we propose to integrate our editing with techniques that predict the reliability of the automatic segmentation [6] to guide the user where editing is needed the most.

**Acknowledgment:** The authors would like to thank Dr. Leo Grady for providing the implementation of the brush editing in [3] and for the valuable discussions and feedback on the development of the splice editing framework.

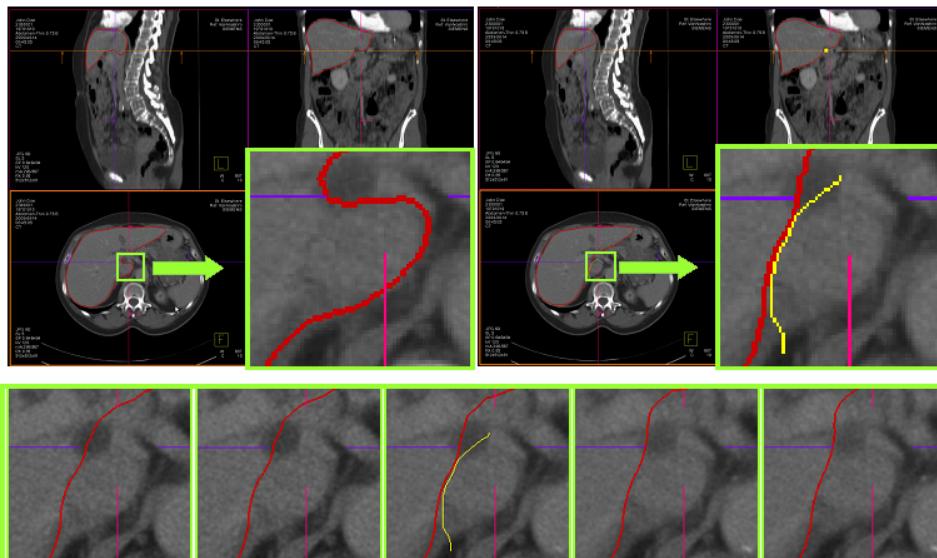
### 5. REFERENCES

- [1] Barrett, W.A., Mortensen, E.N.: Interactive live-wire boundary extraction. *Medical Image Analysis* 1, 331–341 (1997)
- [2] Boykov, Y., Jolly, M.P.: Interactive organ segmentation using graph cuts. In: MICCAI. pp. 276–286 (2000)
- [3] Grady, L., Funka-Lea, G.: An energy minimization approach to the data driven editing of presegmented images/volumes. In: MICCAI 2006. vol. II, pp. 888–895 (Oct 2006)
- [4] Heckel, F., Moltz, J.H., Bornemann, L., et. al.: 3d contour based local manual correction of tumor segmentations in ct scans. In: SPIE (2009)
- [5] O'Rourke, J.: Computational Geometry in C. Cambridge University Press (2011)
- [6] Saad, A., Miller, T., Hamarneh, G.: Probexplorer: Uncertaintyguided exploration and editing of probabilistic medical image segmentation. *Computer Graphics Forum* 29(3)
- [7] Sofka, M., Ralovich, K., Birkbeck, N., Zhang, J., Zhou, S.: Integrated detection network (IDN) for pose and boundary estimation in medical images. In: ISBI. Chicago, IL (2011)
- [8] Yang, H.F., Choe, Y.: An interactive editing framework for electron microscopy image segmentation. In: ISVC (2011)

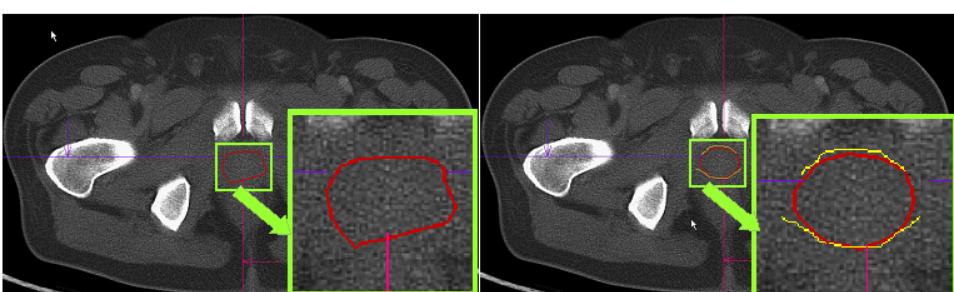
<sup>2</sup>Cropped around the area of interest



(a) Editing of Liver Segmentation in Liver MR Scan



(b) Editing of Liver Segmentation in Liver CT Scan



(c) Editing of Prostate Segmentation in Liver CT Scan



(d) Splice-Guided Editing used as a Virtual Scalpel for Oncology Applications

**Fig. 2.** Splice editing applied to different organs in different modalities.