

Globally optimal coupled surfaces for semi-automatic segmentation of medical images

Juan Eugenio Iglesias

Translational Imaging Group, University College London, UK

Abstract. Manual delineations are of paramount importance in medical imaging, for instance to train supervised methods and evaluate automatic segmentation algorithms. In volumetric images, manually tracing regions of interest is an excruciating process in which much time is wasted labeling neighboring 2D slices that are similar to each other. Here we present a method to compute a set of discrete minimal surfaces whose boundaries are specified by user-provided segmentations on one or more planes. Using this method, the user can for example manually delineate one slice every n and let the algorithm complete the segmentation for the slices in between. Using a discrete framework, this method globally minimizes a cost function that combines a regularizer with a data term based on image intensities, while ensuring that the surfaces do not intersect each other or leave holes in between. While the resulting optimization problem is an integer program and thus NP-hard, we show that the equality constraint matrix is totally unimodular, which enables us to solve the linear program (LP) relaxation instead. We can then capitalize on the existence of efficient LP solvers to compute a globally optimal solution in practical times. Experiments on two different datasets illustrate the superiority of the proposed method over the use of independent, label-wise optimal surfaces ($\sim 5\%$ mean increase in Dice when one every six slices is labeled, with some structures improving up to $\sim 10\%$ in Dice).

1 Introduction

Image segmentation is the process of assigning meaningful labels to pixels (2D) or voxels (3D). In medical imaging, the set of labels corresponds to a number of biologically relevant regions of interest (ROIs), such as different organs, types of cells, etc., as well as a background, in most cases. Segmentation is a key preprocessing step for a wide array of subsequent analyses, such as volumetry or shape analysis, and is therefore one of the core problems in medical imaging.

As useful as traditional manual segmentation is, its usage is encumbered by the necessity that images have to be delineated by an expert, which leads to three main issues: (a) it is not reproducible; (b) it often requires expertise that might not be present at every research center; and (c) it is typically very time consuming – and thus expensive. This last limitation is particularly problematic in 3D data (in which multiple 2D slices are traced to create a 3D segmentation) for two reasons. First, the increasing resolution of medical images requires segmenting a larger number of 2D slices. And second, several iterations are often

needed in order to ensure that segmentations made in a particular orientation are consistent and smooth when displayed in the orthogonal views. Moreover, 3D segmentation is typically a very inefficient process, since a lot of time is spent labeling 2D images that are very similar to their neighboring slices.

Some of the limitations of manual segmentation can be addressed with automated techniques. Most popular families of automatic segmentation methods are supervised and rely on training data, e.g., shape models [1], multi-atlas segmentation [2, 3], probabilistic atlases [4], or voxel classifiers based on learning models such as support vector machines [5], random forests [6] or, more recently, deep neural networks (e.g., [7]). However, creating databases of labeled training data for supervised techniques still requires manually segmenting images. And even for unsupervised methods, manual delineations are needed to produce a gold standard with validation purposes.

A faster, more reproducible way of generating gold standards is through semi-automated segmentation, which represents a compromise between automated and manual methods: the user is required to provide a relatively small amount of input, which an automated algorithm subsequently uses to produce a dense segmentation of the whole image. In 2D, the popular live wire method [8, 9] uses a shortest path algorithm to produce a continuous contour out of a set of points placed along the (possibly ill-defined) boundary of an object. The Grab-Cut algorithm [10] iteratively uses graph cuts [11] to create a segmentation of an object from a rectangular, user-provided bounding box. The Random Walker algorithm [12] uses a set of user-provided scribbles to compute a dense segmentation based on the probability that a random walker starting at each unlabeled pixel first reaches one of the pre-labeled pixels. GeoS [13] also relies on scribbles to produce a geodesic distance map that, in the context of a conditional random field, yields the final segmentation. Many of these methods can be made interactive by allowing the user to modify his input while updating the output in real-time – if this is computationally feasible.

Extending the semi-automated methods described above to 3D is immediate in most cases. A notable exception is live wire, which requires finding a minimal surface joining user-specified contours. Such an extension has a direct, very useful application to semi-automated segmentation: a user can manually delineate one slice every n in a 3D volume (possibly with 2D semi-automated techniques) and then use this method to produce a smart “interpolation” of the segmentation for the unlabeled slices. This strategy effectively exploits the similarity between adjacent slices and can save large amounts of manual labeling effort.

After some attempts in the literature to reconstruct 2D surfaces from sets of 1D curves (e.g., [14]), Grady proposed an algorithm [15] to compute a globally minimal surface given its boundary. The solution is efficiently found by solving a minimum-cost flow problem [16]. Compared with other globally optimal surface algorithms, such as LOGISMOS [17] and its variants (all based on [18]), Grady’s method has the advantages of being able to handle topological changes in the optimal surface; being agnostic to the type and modality of the input images; and not requiring a preliminary volumetric segmentation.

Generalizing Grady’s method to multi-class segmentation is not straightforward. One can apply the algorithm one label at the time and combine the binary outputs into a multi-class segmentation, but this approach has three disadvantages. First, it requires handcrafting rules for handling conflicts (overlaps, holes) when merging segmentations. Second, it is suboptimal in terms of cost function. And third, the quality of the segmentation can in some cases be considerably worse than that of an algorithm that jointly computes the surfaces, e.g., high-contrast ROIs can help the segmentation of neighboring lower-contrast ROIs.

Here we present a generalization of Grady’s algorithm to multiple ROIs. The method computes surfaces simultaneously for all labels while ensuring that no holes or overlaps are produced, and also inherits the ability to handle topological changes in the surfaces. We show that, despite the coupling between the surfaces, the globally optimal solution can still be obtained by solving a linear (rather than integer) program, which can be done in practical times with existing techniques.

2 Methods

2.1 Continuous formulation of joint surface fitting

In the continuous domain, the problem of joint surface fitting can be framed as a constrained optimization problem, illustrated in Fig. 1. Given a cuboid shaped 3D image domain Ω , let $\{\mathcal{U}_l : l = 1, \dots, L\}$ represent user-provided segmentations for L different ROIs (including the background), defined on one or more planes. In our target application, there would be two parallel planes¹, but this does not need to be the case in our formulation. Let $\{\mathcal{R}_l = \delta\mathcal{U}_l : l = 1, \dots, L\}$ represent the boundaries of the segmentations; note that each \mathcal{R}_l can represent more than one contour, e.g., contours on different planes, as in Fig. 1. Finally, let $\{\mathcal{Z}_l : l = 1, \dots, L\}$ represent the L surfaces to fit; note that \mathcal{Z}_l is necessarily an open surface since it has a non-empty boundary \mathcal{R}_l . The problem is then:

$$\begin{aligned} \min_{\{\mathcal{Z}_l\}} \quad & \sum_{l=1}^L \left(\alpha \int_{\mathcal{Z}_l} dS + \int_{\mathcal{Z}_l} g_l(I, \hat{\mathbf{n}}; \beta) dS \right) \\ \text{s.t.} \quad & \mathcal{B}(\mathcal{Z}_l) = \mathcal{R}_l, \quad \bigcup_{l=1}^L \text{Vol}(\mathcal{Z}_l \cup \mathcal{U}_l) = \Omega, \quad \bigcap_{l=\{1,2\}} \text{Vol}(\mathcal{Z}_l \cup \mathcal{U}_l) = \emptyset, \forall l_1 \neq l_2, \end{aligned} \quad (1)$$

where dS is the area element; \mathcal{B} is the boundary operator; $\text{Vol}(\mathcal{Z}_l \cup \mathcal{U}_l)$ is the 3D segmentation enclosed by $\mathcal{Z}_l \cup \mathcal{U}_l$; α, β are scalar constants; I represents the image intensities; and $\hat{\mathbf{n}}$ is the surface normal at each point (pointing towards the inside of the volume). The first term in Eq. 1 is a regularizer that penalizes the total surface area with relative weight α . The second term encourages the surfaces to follow perpendicular image gradients, by means of a function g_l of

¹ If there are more than two segmented slices, we can solve for one gap between labeled planes at the time because the boundary conditions decouple the problems.

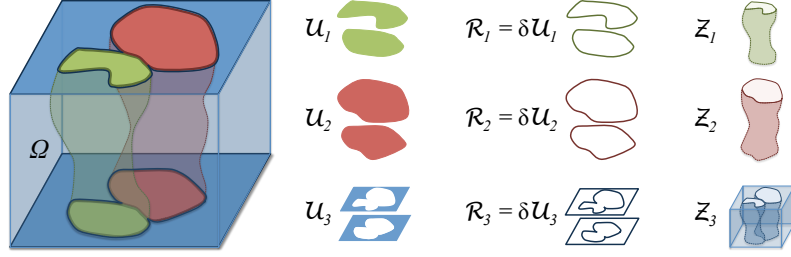


Fig. 1. We aim to find the open surfaces $\{Z_l\}$ that: (a) globally minimize the cost function in Eq. 1; (b) are constrained to have boundaries $\{R_l\}$ given by the contours of user-specified segmentations $\{U_l\}$; and (c) do not intersect or leave holes in between.

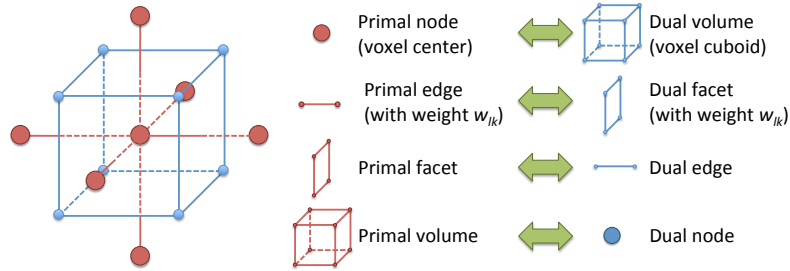


Fig. 2. Duality in 3D, 6-connected lattice. The image intensities are defined on the primal nodes. The surfaces we consider in this paper are given by sets of dual facets, which are equivalent to the faces of the cuboids representing image voxels.

the image intensities and the normal $\hat{\mathbf{n}}$, parametrized by β . A frequent choice is:

$$g_l(I, \hat{\mathbf{n}}; \beta) = \exp(-\beta \|\nabla_{\hat{\mathbf{n}}} I\|^2), \quad (2)$$

which we use in this study. Note that the cost of a surface is independent of its label l in Eq. 2, but this does not need to be the case in general.

2.2 Discrete formulation

We follow [15] to discretize the problem in Eq. 1, which enables us to compute a globally optimal solution. We assume that our image intensities are defined at the center of cuboid voxels. These centers correspond to the nodes of a 6-connected 3D lattice, which defines our primal graph (see Fig. 2). The (primal) nodes are connected by K primal² edges $\{e_k^p : k = 1, \dots, K\}$, which have corresponding (label-dependent) weights $\mathbf{w}_l = [w_{l1}, \dots, w_{lK}]^T$. In addition, we also consider the dual graph, in which nodes corresponds to volumes, edges to facets, and vice versa. The volumes of the dual graph correspond to the cuboids of the image voxels, such that the dual facets $\{f_k^d : k = 1, \dots, K\}$ correspond to the

² We use superscript p for primal and superscript d for dual.

faces of these cuboids (where our surfaces will be defined). The dual facets have a direct correspondence with the primal edges, and hence share the same weights w_{lk} . Note that it is crucial to include each primal edge / dual facet twice in $\{e_k^p\}$, $\{f_k^d\}$ and \mathbf{w}_l with opposite orientations. This is because we need to discern whether the segmented ROI is on one side of the dual facet or the other, following the convention that the surface normal points toward the inside of the enclosed volume. In a similar fashion, we also consider the M dual edges $\{e_m^d : m = 1, \dots, M\}$ with corresponding primal facets $\{f_m^p : m = 1, \dots, M\}$.

To represent each surface, we use a discrete function $\mathbf{z}_l = [z_{l1}, \dots, z_{lK}]^T$ that encodes whether dual facet f_k^d is part of the l^{th} surface (or, equivalently, whether primal edge e_k^p intersects the surface), and with what multiplicity. Hence, we are replacing each continuous surface \mathcal{Z}_l by a set of voxel faces represented by \mathbf{z}_l .

Cost function: We discretize the cost function by replacing integrals by sums in Eq. 1. If dual facet f_k^d is part of the l^{th} surface, its contribution to the cost is:

$$w_{lk} = \text{Area}(f_k^d) (\alpha + \exp[-\beta(I_{j_1} - I_{j_2})^2]),$$

where I_{j_1} and I_{j_2} are the image intensities at the primal nodes defining the corresponding primal edge e_k^p ; and $\text{Area}(f_k^d)$ is the area of facet f_k^d , given by the voxel dimensions. In our case, due to the choice of g_l in Eq. 2, the weight is the same for both orientations of the edge, but this does not need to be the case in general. We have kept the weights $\{\mathbf{w}_l\}$ separated in the notation (even if they are the same for all ROIs) for practical reasons in the implementation of the algorithm (Section 3.2 below). The total cost to minimize is then given by:

$$\min_{\{\mathbf{z}_l\}} \sum_l \mathbf{w}_l^T \mathbf{z}_l = \sum_l \sum_k w_{lk} z_{lk}. \quad (3)$$

Constraints: We need to consider two sets of constraints. First, we must ensure that the boundaries of the surfaces correspond to the contours of the specified 2D segmentations. This can be achieved through the edge-facet incidence matrix of the dual graph \mathbf{B} , which yields the discrete 3D boundary operator:

$$B_{mk} = \begin{cases} 1, & \text{if } e_m^d \text{ borders } f_k^d \text{ with coherent orientation,} \\ -1, & \text{if } e_m^d \text{ borders } f_k^d \text{ without coherent orientation,} \\ 0, & \text{otherwise,} \end{cases} \quad (4)$$

where orientation coherence is given by the right-hand rule. Next, we define the contour vectors $\{\mathbf{r}_l : l = 1, \dots, L\}$, also on the dual graph, which represent the boundaries of the user-defined segmentations on one or more slices:

$$r_{lm} = \begin{cases} 1, & \text{if } e_m^d \text{ is on the } l^{\text{th}} \text{ contour with coherent orientation,} \\ -1, & \text{if } e_m^d \text{ is on the } l^{\text{th}} \text{ contour without coherent orientation,} \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

With these definitions of \mathbf{B} and \mathbf{r}_l , we can write the constraints simply as:

$$\mathbf{B}\mathbf{z}_l = \mathbf{r}_l, \quad \forall l. \quad (6)$$

The second set of constraint specifies that the union of the enclosed volumes needs to be equal to the image domain Ω , while the intersection must be the empty set. We define the matrix \mathbf{C} and vector \mathbf{t} as:

$$C_{kk'} = \begin{cases} 1, & \text{if } k' = k, \\ -1, & \text{if } k' = \tilde{k}(k), \\ 0, & \text{otherwise,} \end{cases} \quad t_k = \begin{cases} 1, & \text{if } f_k^d \in \delta\Omega_{[\text{inwards}]}, \\ -1, & \text{if } f_k^d \in \delta\Omega_{[\text{outwards}]}, \\ 0, & \text{otherwise,} \end{cases} \quad (7)$$

where $\tilde{k}(k)$ is the index of facet f_k^d with opposite orientation to f_k^d , whereas $\delta\Omega_{[\text{inwards}]}$ and $\delta\Omega_{[\text{outwards}]}$ are the sets of dual facets on the external boundary of Ω whose associated normals point towards the inside and outside of the image domain, respectively. We also define $\mathbf{C}^* = [\mathbf{C} \ \mathbf{C} \cdots \mathbf{C}]$, and $\mathbf{z}^* = [z_1^T \cdots z_L^T]^T$. Then, the constraints can be encoded in the following system of linear equations:

$$\mathbf{C}^* \mathbf{z}^* = \mathbf{t}. \quad (8)$$

Forcing $\sum_{l=1}^L (z_{lk} - z_{l\tilde{k}})$ in Eq. 8 to be zero inside Ω ensures that, if a facet is part of a surface on one side, it must also be part of another on the other side, thus avoiding holes and overlaps in the segmentation. Forcing the sum to be ± 1 (depending on the orientation) on the walls ensures that the whole image domain will be covered by the surfaces. In practice, including one of the orientations in \mathbf{C} and \mathbf{t} is sufficient, since swapping k and \tilde{k} yields the same constraints.

2.3 Integer program and linear programming relaxation

We can combine the cost function in Eq. 3 with the constraints defined in Eq. 6 and 8 to define the following integer program (IP):

$$\begin{aligned} \min_{\mathbf{z}^*} \quad & \mathbf{W}^T \mathbf{z}^* \\ \text{s.t.} \quad & \mathbf{A} \mathbf{z}^* = \mathbf{v} \\ & \mathbf{z}^* \succeq 0, \\ & \mathbf{z}^* \in \mathbb{Z}^{L \times K}, \end{aligned} \quad (9)$$

where we have defined:

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_L \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} \mathbf{B} & 0 & \cdots & 0 \\ 0 & \mathbf{B} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{B} \\ \hline & & & \mathbf{C}^* \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \vdots \\ \mathbf{r}_L \\ \mathbf{t} \end{bmatrix}. \quad (10)$$

Integer programming is notoriously an NP-hard problem [19]. Therefore, solving (9) directly is impractical. However, it can be shown [20] that, if the equality constraint matrix \mathbf{A} is totally unimodular³ (TU) and the equality constraint

³ A matrix is TU if it is integer and every square submatrix has determinant 1 or -1.

vector \mathbf{v} is integer valued, then the linear programming (LP) relaxation of the IP is guaranteed to produce an integer solution, which is equal to the solution of the original IP (note that this condition is sufficient, but not necessary).

Lemma. *The equality constraint matrix \mathbf{A} in IP (9) is TU.*

Proof. It can be shown [21] that a sufficient condition for TU is that the columns of the matrix add up to zero while all its elements are in the set $\{-1, 0, 1\}$. Here, we first observe that the matrix \mathbf{B} can be partitioned into $\mathbf{B} = [\mathbf{B}_1 | \mathbf{B}_2]$, such that \mathbf{B}_1 includes each dual facet only once – no matter with which of the two possible orientations. Then, if we sort the columns of \mathbf{B}_2 such that they follow the same facet order as \mathbf{B}_1 , we have that $\mathbf{B}_2 = -\mathbf{B}_1$, since the columns correspond to the same facets but with opposite edge orientations. Hence, the columns of \mathbf{B} add up to the zero vector. The columns of \mathbf{C}^* also add up to the zero vector, since each row contains L elements equal to 1 and L equal to -1 . Therefore, the columns of \mathbf{A} also add up to zero (Eq. 10). Since all elements of \mathbf{A} are in the set $\{-1, 0, 1\}$ (from the definitions of \mathbf{B} and \mathbf{C}^* in Eqs. 4 and 7), then \mathbf{A} is TU.

Since \mathbf{A} is TU and \mathbf{v} is integer (see the definitions of \mathbf{r}_i and \mathbf{t} in Eqs. 5 and 7), the following LP relaxation yields the same solution as (9):

$$\begin{aligned} \min_{\mathbf{z}^*} \quad & \mathbf{W}^T \mathbf{z}^* \\ \text{s.t.} \quad & \mathbf{A} \mathbf{z}^* = \mathbf{v} \\ & \mathbf{z}^* \succeq 0, \\ & \mathbf{z}^* \in \mathbb{R}^{L \times K}, \end{aligned} \tag{11}$$

Linear programming has polynomial complexity, and efficient solvers exist to compute the globally optimal solution of (11) in practical times.

3 Experiments and results

3.1 Data

We used two publicly available datasets in our experiments. The first dataset consists of 35 T1-weighted brain MRI scans from the 2013 MICCAI SATA challenge⁴. The images were acquired on a 3T scanner with an MP-RAGE sequence at 1 mm isotropic resolution. Fourteen structures were labeled by experts in coronal plane: left and right amygdala, caudate, accumbens, hippocampus, putamen, thalamus and pallidum. We refer to these data as the “brain dataset”.

The second dataset consists of five *ex vivo* MRI scans of single human hippocampi (3 right, 3 left) [22]. The scans were acquired on a 9.4T scanner with a multi-spin echo sequence at $0.2 \times 0.2 \times 0.3$ mm resolution (coronal). Five hippocampal subfields were manually delineated on the images by an expert: CA1; CA2 and CA3 (CA23); hilus of the dentate gyrus (DG:H); stratum radiatum, stratum lacunosum-moleculare and the vestigial hippocampal sulcus

⁴ <https://www.synapse.org/#!/Synapse:syn3193805/wiki/217780>

(SR+SLM+VHS); and stratum moleculare of the dentate gyrus (DG:SM). The manual segmentations were mostly made on the coronal plane, with verification in the sagittal plane. We refer to these images as the ‘‘hippocampal dataset’’.

3.2 Experimental setup

We evaluated the performance of our algorithm as a function of n_{skip} , the number of unlabeled slices between each pair of labeled slices. As baseline approach, we applied Grady’s algorithm to each label independently, and then merged the resulting L binary segmentations into a multi-label volume – if two segmentations overlapped in given region, we selected the label that gave the lowest cost (Eq. 3).

For each value of n_{skip} , we randomly sampled 10 and 50 stacks of $n_{\text{skip}} + 2$ coronal slices from each scan of the brain and hippocampal datasets, respectively. This yielded 600 test volumes (350 brain, 250 hippocampal), in which the segmentation was assumed to be known for the first and last slice. We then used the two competing methods to segment the rest of the slices in each volume, and measured the overlap with the ground truth using Dice scores. We merged the results of the left and right side of each ROI for simplicity of presentation.

Implementation details: We min-max normalized the images to $[0, 255]$, and manually tuned the parameters (on a separate brain MRI dataset) to $\beta = 0.005$ and $\alpha = 10^{-5}/n_{\text{skip}}$. We solved the LP (11) with Gurobi 7.0 (www.gurobi.com). To handle the boundaries $\delta\Omega$ for which no segmentation was specified by the user (e.g., the ‘‘walls’’ completing the surface of the cuboid between two labeled slices), we assumed that the background ROI surrounded all others (which was the case in all our experiments) and set the costs w_{lk} of the corresponding surface facets (with the normal facing inwards) to a large negative constant:

$$\mathbf{W} = \begin{bmatrix} \tilde{w}_1 \\ \vdots \\ \tilde{w}_L \end{bmatrix}, \text{ where } \tilde{w}_{lk} = \begin{cases} -\infty, & \text{if } l = \text{background, } f_k^d \in \delta\Omega_{[\text{inwards}]} \text{ and } f_k^d \\ & \text{does not correspond to a segmented voxel} \\ w_{lk}, & \text{otherwise.} \end{cases}$$

This setup also enabled us to drop the rows corresponding to the dual facets on $\delta\Omega$ from the \mathbf{C} matrix (Eq. 7), speeding up convergence of the solver in practice.

3.3 Results

Figures 3 and 4 show the results for the brain and hippocampal datasets, respectively, whereas Figs. 5 and 6 display sample outputs for the two competing algorithms. In the brain dataset, the proposed method exploits the neighboring relationships between the structures in order to outperform the baseline algorithm for every ROI. The differences between the two methods become larger as the separation n_{skip} grows, averaging 5% Dice at $n_{\text{skip}} = 5$. The gap is particularly large ($>11\%$ at $n_{\text{skip}} = 5$) for the pallidum; this structure is often heavily undersegmented when processed on its own, but frequently emerges when segmented jointly with the neighboring, high-contrast putamen (as in Fig. 5).

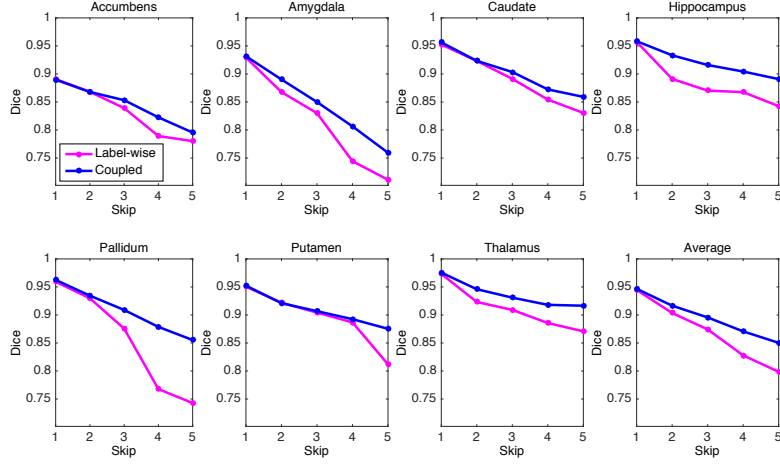


Fig. 3. Brain dataset: Dice score as a function of the number of unlabeled slices between manually segmented slices for each brain structure, as well as average across structures.

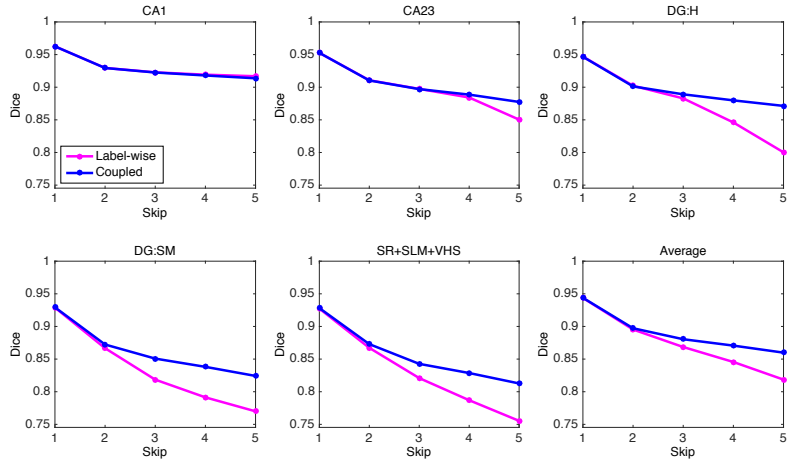


Fig. 4. Hippocampal dataset: Dice score as a function of the number of unlabeled slices between manually segmented slices for each subfield, as well as average across subfields.

The results are similar in the hippocampal dataset. For the larger, high-contrast CA1, both methods produce almost identical results. However, for the internal, lower-contrast subfields, the proposed method outperforms the baseline, averaging 4% higher Dice by $n_{\text{skip}} = 5$. The difference is largest for the stratum moleculare (DG:SM) and hilus (DG:H) of the dentate gyrus, which share a boundary that is practically invisible in the images (see example in Fig. 6).

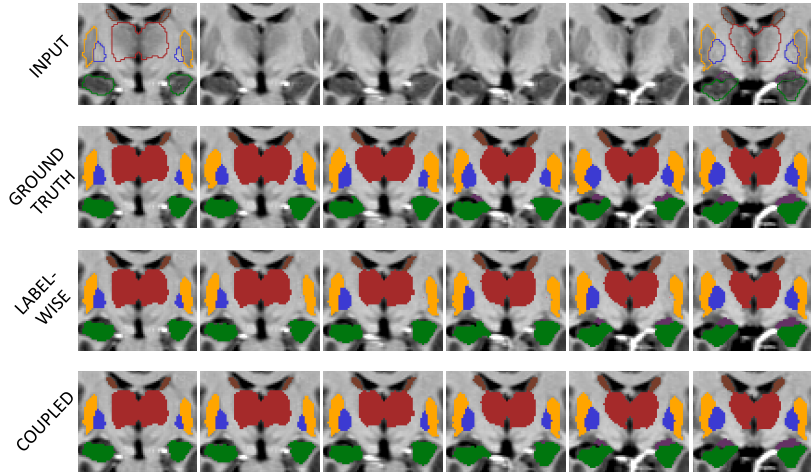


Fig. 5. Sample segmentations from brain dataset with $n_{\text{skip}} = 4$. Green is hippocampus, orange is putamen, blue is pallidum, dark red is thalamus, brown is caudate, and purple is amygdala. Note the missing pallidum in the baseline approach.

In absolute terms, the proposed algorithm yields satisfactory results (Dice above 85-90%) for many structures all the way to $n_{\text{skip}} = 5$ (hippocampus, putamen, thalamus, CA1, CA23, DG:H), and for most structures at $n_{\text{skip}} \leq 3$. Exceptions are ROIs with faint boundaries and poor reliability (e.g., amygdala, SR+SLM+VHS). However, even low values of n_{skip} can be very useful in practice, since they can save $100 \times n_{\text{skip}} / (1 + n_{\text{skip}})$ percent of manual labeling effort.

4 Conclusion

We have presented a semi-automated segmentation method that can compute a globally optimal set of discrete coupled surfaces, whose boundaries are specified by the contours of user-provided delineations on one or more (typically parallel) slices. The results have shown that the method outperforms the application of the binary version to each ROI independently.

The proposed method can easily be made interactive: if the segmentation is not satisfactory, feedback can be provided to correct the output. More specifically, the user can mark an oriented boundary between two voxels, specifying which label l should be found on a given side of it. Then, the weight of the facet at hand w_{lk} (with the appropriate orientation) can be set to a large, negative constant. This procedure effectively forces the surfaces to go through the specified points when the LP in (11) is solved to update the segmentation – since the solver is guaranteed to yield the global optimum.

In addition to the difficulties to handle surfaces with high curvature (inherited from [15]), the main limitation of the method is its computational requirements. The null-space of \mathbf{A} in Eq. 10 does not yield a concatenation of volume-facet

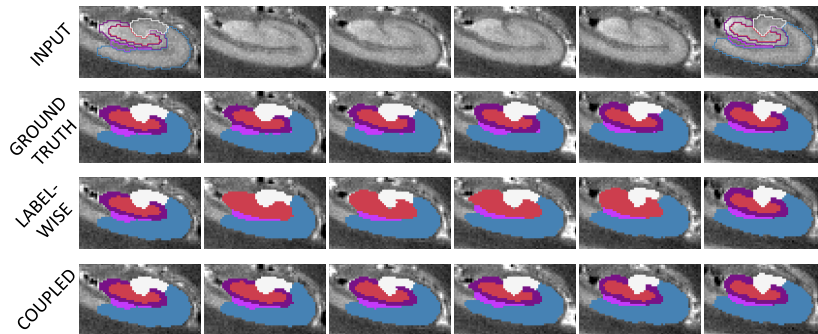


Fig. 6. Sample segmentations from the hippocampal dataset with $n_{\text{skip}} = 4$. Blue is CA1, white is CA23, red is DG:H, violet is DG:SM and pink is SR+SLM+VHS. Note the missing DG:SM in the baseline method.

incidence matrices due to the coupling terms introduced by Eq. 8. This prevents rewriting the optimization as a minimum-cost flow problem, which can be more efficiently solved [15]. We were able to solve the LP relaxation relatively quickly in our experiments (minutes in the worst cases), but computation times grow quickly with image size. This is not a problem when “interpolating” segmentations across slices (a task that can be carried out offline), but it is a limiting factor for the interactive extension of the algorithm discussed above.

Future work will follow four directions. First, we will investigate ways of further simplifying the LP relaxation (Eq. 11). If we identify the graph for which \mathbf{A} is the incidence matrix, we can devise a faster optimization algorithm based on its dual graph. Second, we will implement and validate the interactive version of the algorithm. Third, we will evaluate the algorithm more extensively, including experiments with multiple orientations (e.g., 2.5D) and comparisons with other methods (e.g., alpha-expansion [11]). And fourth, we will explore other possible definitions of g_l (Eq. 2); while a simple exponential captures most visible edges (relying on the regularizer around ill-defined boundaries), ROI-specific weights computed with supervised edge detectors should yield higher performance.

As the amounts of available imaging data and their resolution grow, and as the requirements of labeled data to train state-of-the-art supervised algorithms increase, we believe that semi-automated approaches like the method proposed in this paper will become increasingly important in medical image analysis.

Acknowledgement: This research was supported by the European Research Council (Starting Grant 677697, project BUNGEE-TOOLS), and would not have been possible without the suggestions from Dr. Leo Grady (HeartFlow, Inc.).

References

1. Cootes, T., Taylor, C., Cooper, D., Graham, J.: Active shape models—their training and application. *Computer Vision and Image Understanding* **61**(1) (1995) 38–59

2. Rohlfing, T., Brandt, R., Menzel, R., Maurer, C.R.: Evaluation of atlas selection strategies for atlas-based image segmentation with application to confocal microscopy images of bee brains. *NeuroImage* **21**(4) (2004) 1428–1442
3. Iglesias, J.E., Sabuncu, M.R.: Multi-atlas segmentation of biomedical images: a survey. *Medical Image Analysis* **24**(1) (2015) 205–219
4. Ashburner, J., Friston, K.J.: Unified segmentation. *NeuroImage* **26** (2005) 839–851
5. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: *Proceedings of the 5th annual workshop on Computational learning theory*. (1992) 144–152
6. Breiman, L.: Random forests. *Machine learning* **45**(1) (2001) 5–32
7. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D U-Net: Learning dense volumetric segmentation from sparse annotation. In: *Proceedings of Medical Image Computing and Computer-Assisted Intervention*. (2016) 424–432
8. Falcão, A.X., Udupa, J.K., Samarasekera, S., Sharma, S., Hirsch, B.E., Lotufo, R.d.A.: User-steered image segmentation paradigms: Live wire and live lane. *Graphical Models and Image Processing* **60**(4) (1998) 233–260
9. Mortensen, E.N., Barrett, W.A.: Interactive segmentation with intelligent scissors. *Graphical Models and Image Processing* **60**(5) (1998) 349–384
10. Rother, C., Kolmogorov, V., Blake, A.: GrabCut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics* **23**(3) (2004) 309–314
11. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23**(11) (2001) 1222–1239
12. Grady, L.: Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28**(11) (2006) 1768–1783
13. Criminisi, A., Sharp, T., Blake, A.: GeoS: Geodesic image segmentation. In: *Proceedings of European Conference on Computer Vision*, Springer (2008) 99–112
14. Falcão, A.X., Udupa, J.K.: A 3D generalization of user-steered live-wire segmentation. *Medical Image Analysis* **4**(4) (2000) 389–402
15. Grady, L.: Minimal surfaces extend shortest path segmentation methods to 3D. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **32**(2) (2010) 321–334
16. Sullivan, J.M.: A crystalline approximation theorem for hypersurfaces. PhD thesis, Princeton (1990)
17. Yin, Y., Zhang, X., Williams, R., Wu, X., Anderson, D.D., Sonka, M.: LOGIS-MOS – layered optimal graph image segmentation of multiple objects and surfaces: cartilage segmentation in the knee joint. *IEEE Transactions on Medical Imaging* **29**(12) (2010) 2023–2037
18. Li, K., Wu, X., Chen, D.Z., Sonka, M.: Optimal surface segmentation in volumetric images—a graph-theoretic approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28**(1) (2006) 119–134
19. Papadimitriou, C.H.: On the complexity of integer programming. *Journal of the ACM* **28**(4) (1981) 765–768
20. Grady, L.J., Polimeni, J.: *Discrete calculus: Applied analysis on graphs for computational science*. Springer Science & Business Media (2010)
21. Schrijver, A.: *Theory of linear and integer programming*. John Wiley & Sons (1998)
22. Yushkevich, P.A., Avants, B.B., Pluta, J., Das, S., Minkoff, D., Mechanic-Hamilton, D., Glynn, S., Pickup, S., Liu, W., Gee, J.C., Grossman, M., Detreb, J.: A high-resolution computational atlas of the human hippocampus from postmortem magnetic resonance imaging at 9.4T. *NeuroImage* **44**(2) (2009) 385–398