Eurographics Conference on Visualization (EuroVis) 2017
J. Heer, T. Ropinski and J. van Wijk
(Guest Editors)

Volume 36 (2017), Number 3

Visualizing the Uncertainty of Graph-based 2D Segmentation with Min-path Stability

B. Summa[1], J. Tierny[2], and V. Pascucci[3]

[1]Tulane University
[2]Sorbonne Universites, UPMC Univ Paris 06, CNRS, LIP6 UMR 7606
[3]Scientific Computing and Imaging Institute, University of Utah

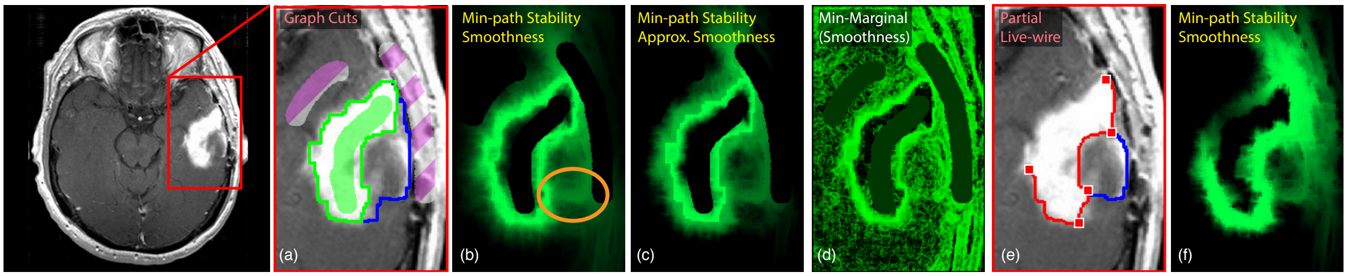

Figure 1: *Segmentation of an MRI scan of a malignant tumor in the right hemisphere of the brain. (a) Graph cuts segmentation of the tumor with annotation to denote tumor (green) and not tumor (purple striped). The segmentation (green contour) misses important tumor structure. Our alternative path extraction finds a valid alternative segmentation (blue) that includes this structure. ($\beta = 0.6$) (b) The uncertainty in the smoothness term that encodes both the noise and blur in the data and also the ridge of the alternate path. (c) Our fast approximation to the smoothness stability that still finds all salient uncertainty features. (d) This structure is lost in the min-marginal visualization [KT06]. (e) A live-wire, partial segmentation of the tumor with an alternative path (blue) on the middle segment ($\beta = 0.1$) (f) Our uncertainty visualization can be applied to both graph cuts and live-wire segmentations, even partial live-wire segmentations as illustrated. Image courtesy of Kyrre E. Emblem et al [Eea13].*

Abstract

This paper presents a novel approach to visualize the uncertainty in graph-based segmentations of scalar data. Segmentation of 2D scalar data has wide application in a variety of scientific and medical domains. Typically, a segmentation is presented as a single unambiguous boundary although the solution is often uncertain due to noise or blur in the underlying data as well as imprecision in user input. Our approach provides insight into this uncertainty by computing the "min-path stability", *a scalar measure analyzing the stability of the segmentation given a set of input constraints. Our approach is efficient, easy to compute, and can be generally applied to either graph cuts or live-wire (even partial) segmentations. In addition to its general applicability, our new approach to graph cuts uncertainty visualization improves on the time complexity of the current state-of-the-art with an additional fast approximate solution. We also introduce a novel query enabled by our approach which provides users with alternate segmentations by efficiently extracting local minima of the segmentation optimization. Finally, we evaluate our approach and demonstrate its utility on data from scientific and medical applications.*

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Methodology and Techniques—Graphics data structures and data types : I.4.6 [Image Processing and Computer Vision]: Segmentation—Pixel classification

1. Introduction

A driving force in visualization is the effective and faithful communication of data. It is then of no surprise that visualization of uncertainty in the stages of a scientific workflow has emerged as a critical research area. In terms of workflow components, segmentation of scalar field data, in particular, has become an indispensable tool. It is a fundamental element for applications in areas such as geosciences [HBG*11], material sciences [WZS*13], biological sciences [JM10, BDVJ*16], and medical imagery [FLBF*06, DLD*15, KMW*15] to name a few. For example, geoscientists extract horizons from seismic tomography for oil exploration; neuroscientists segment neurons in high resolution microscopy scans to gain a deeper understanding of the brain; and doctors or technicians segment tumors from patient fMRI scans to plan surgery or treatment regimens.

Graph-based approaches comprise a popular set of algorithms for data segmentation. Graph cuts [BVZ01, BK04] and livewire [MB98] are two such interactive techniques that are found in a large number of data manipulation environments. These approaches produce segmentations as the minimization of an energy function, yielding a solution where every data point is given a single categorization. However, the presence of noise or poor sampling translates to uncertainty in a segmentation. Moreover, slight perturbations of user input can yield drastically different segmentations if in areas of instability. Single-categorization approaches hide this uncertainty from a user, which can bias the insight derived from visual analysis and lead to significant errors in judgment. For example, based on a segmentation a surgeon may conclude that a distinct tumor boundary exists despite noisy data, leading to poor margins for excision. In neurobiology, single solutions can hide alternate interpretations, which can lead to erroneous nerve connections in neurological pathway mapping.

In this work, we present a new scalar measure, called the *"min-path stability"*, which depicts the required cost to shift a segmentation to another local minima of the segmentation optimization energy. This measure highlights ambiguous configurations where multiple, sometimes highly distinct, data segmentations are possible while only being separated by a small energy difference. In addition to providing fast algorithms for the computation of this measure, we also present an approach to easily query these alternate segmentations. We show with practical data-sets how our measure can provide key insights into the uncertainty of the segmentation due to, for instance, input noise or imprecision in user inputs.

This paper makes the following new contributions:

- A novel and unified uncertainty measure for graph cuts and livewire segmentations;
- An algorithm for uncertainty evaluation in graph cuts segmentations with lower time complexity than state-of-the-art;
- A novel dynamic shortest path tree algorithm that provides further practical speed-ups for the algorithm mentioned above;
- A fast, approximate uncertainty visualization for graph cuts segmentations; and
- A fast algorithm to extract locally minimum distinct alternate segmentations

2. Related Work

Visualization of uncertainty has emerged as a critical research area, especially in scientific workflows [BHJ*14]. For example, there has been work in quantification or visualization of uncertainty in an ensemble of segmentations [WZW04, ATHL14b] or isocontours [PWH11, WMK13]. Rather than an ensemble, our work visualizes the uncertainty of a segmentation given a single set of user constraints. There has been work on visualization of uncertainty in graphs, but their primary focus has been on less structured graph networks [CCP07, LRCP07].

Early work on quantifying the uncertainty in data segmentation dealt with combining region growing segmentation with edge detection to determine where there may be poor segmentation boundaries [Del91]. There has been work on using entropic thresholding [HPD95, SS04] to encode uncertainty for pixel-wise energy. There has also been work in the visualization of uncertainty in segmentation in the presence of shape and appearance priors [SHM10]. In our work we target more general graph segmentations without prior knowledge.

For more general segmentations, there has been work in producing uncertainty visualizations [SMH10] from the results of segmentations using a Gaussian mixture model [ZBS01], random path walking [Gra06], or min-marginal energies [KT06, TA12]. Praßni et al. [PRH10] have shown how the visualization of a probabilistic field generated by a segmentation from Grady's random walker [Gra06] can provide uncertainty information in a volume segmentation. While graph cuts and the random walker approach can be considered as optimizations in a generalized framework [SG07], each targets a different optimization (ℓ_1 and ℓ_2 respectively). Therefore, these two approaches will produce different segmentations. In addition, Grady's random walker requires a free parameter (called β) that has a significant effect on the segmentation produced. Given these two core differences, the approach by Praßni et al. [PRH10] cannot be used generally as a stability metric for graph cuts. Illustrations of such differences are provided in Fig. 10. Additional approaches [PGA13, ATHL14a] have visualized a measure of the uncertainty of probabilistic segmentations based on entropy or Kullback-Leibler divergence. Similarly, these measures are not directly applicable to graph-cut or live-wire segmentations, which are not probabilistic.

Approaches for interactive data segmentation and editing [HBS*12, HMTH13] often introduce uncertainty assessments. This is the case for the software TurtleSeg, which provides stability metrics in their Spotlight tool [THA10] for live-wire segmentations. As we will detail, in our work we provide a general approach for both live-wire and graph cuts segmentations. The live-wire portion of our technique employs a similar concept to TurtleSeg's stability field but with two major differences. First, our target is the visualization of the stability while their technique uses the stability field in conjunction with other fields to provide automatic user guidance. Second, their stability field is based on the euclidean distance of the path geometry from a single edge perturbation while our field is the cost based on the energy function. Our work therefore encodes where an alternative segmentation would likely follow, not the instability at a point in the field. Previous work has constructed a statistically diverse set of graph-cuts segmentations [BYGRS12] with a greedy algorithm. In contrast, our alternative segmentations are directly obtained as by-products of our stability measure. Also, they can be generally applied to livewire and graph cuts.

Of the previous work, the most apt comparison to our work is that of Kohli and Torr [KT06] who use min-marginal energies and successive application of graph cuts to produce their measure of uncertainty in a graph cut segmentation. As we will show, our work is a complementary approach. In particular, the min-marginal energy is often over-sensitive to the presence of noise and it can result in an overwhelming visualization. Our approach will encode how perturbations in the segmentation are effected by noise, leading to a more intuitive visualization. In addition, our work can be applied generally to either graph cuts or live-wire and our worst case time complexity is lower with an additional fast approximation.

3. Background

To perform a segmentation, a graph is first constructed, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the nodes $\mathcal{V}$ correspond to the pixels in the input scalar field, p, and the edges $\mathcal{E}$ encode all pixel neighbors, (p,q). Segmentations are then a partition or cut of this graph that induces a labeling of all nodes in the graph, L. We denote the label of a node p as l_p.

Graph Cuts Graph-based segmentation algorithms like graph cuts produce this labeling by minimizing an objective energy:

$$E(L) = \sum_{p \in \mathcal{V}} E_d(p, l_p) + \sum_{(p,q) \in \mathcal{E}} E_s(p, l_p, q, l_q). \tag{1}$$

In Eq. 1, E_d is called the *data*, or likelihood, energy and gives the cost associated with giving p label l_p. E_s is called the *smoothness*, or prior, energy and denotes the cost of neighbors p and q having labels l_p and l_q, respectively. In order for the *smoothness* energy to be well-posed, initial label constraints are required either from a user or by an automated approach to enforce labelings for certain nodes.

When the number of possible labels in L is greater than 2, minimizing the energy has been shown to be NP-hard. Therefore, algorithms such as graph cuts producing a k-labeling, where $k > 2$, are an approximate solution. In this work, we target a visualization that illustrates the stability of a segmentation only due to the data and a user's input. To this, we have chosen to target binary ($\mathcal{F}$oreground/ $\mathcal{B}$ackground) segmentation for which the exact solution can be computed without error added by the underlying algorithm. For binary segmentations, graph cuts is a minimum cut of the graph but for clarity we adopt the common name, *graph cuts* (not to be confused with spectral graph-cuts [SK13]). As discussed by Praßni et al. [PRH10], binary segmentations are particularly useful in practice, especially for tasks where users isolate features in an image. Also, as suggested by the same authors, a binary approach can still be used to iteratively segment multiple features.

The data energy term, E_d, can be any per-pixel, per-label function, but is typically based on a distance between a pixel color and some target color for foreground and background. For this work, we will use the data energy of the work of Li et al. [LSTS04]. We refer the reader to this work for more details. Given sets of foreground and background pixels defined by the initial constraints, $\{\mathcal{C}_n^{\mathcal{F}}\}$ and $\{\mathcal{C}_m^{\mathcal{B}}\}$ respectively, for every pixel the distances to these sets are defined as $d_p^{\mathcal{F}} = \min_n \|\mathcal{I}_p - \mathcal{C}_n^{\mathcal{F}}\|$ and $d_p^{\mathcal{B}} = \min_m \|\mathcal{I}_p - \mathcal{C}_m^{\mathcal{B}}\|$ where $\mathcal{I}_p$ is the intensity or color of pixel p. E_d is defined as:

$$E_d(p, l_p) = \begin{cases} d_p^{\mathcal{F}} / (d_p^{\mathcal{F}} + d_p^{\mathcal{B}}) & l_p = \mathcal{F} \\ d_p^{\mathcal{B}} / (d_p^{\mathcal{F}} + d_p^{\mathcal{B}}) & l_p = \mathcal{B} \end{cases} \tag{2}$$

for nodes that are unconstrained in application of foreground or background labels. For constrained pixels, E_d is 0 for the applied label constraint and ∞ for other labels. The data energy is stored on edges from our pixel nodes to two logical source, s, and target, t, nodes used for the cut. Each logical node is associated with either the foreground or background labelings (Fig. 2 (a)).

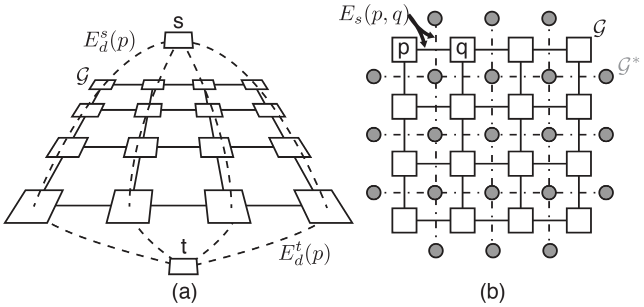

Figure 2: *(a) A pixel graph, $\mathcal{G}$ used for graph cuts segmentation. The nodes in the graph can have a per node per label data energy, E_d. These are encoded in the edges to logical source, s, and target, t, nodes in the graph. (b) $\mathcal{G}$ can also have edges weighted with the smoothness energy, E_s. For a planar graph, a dual graph $\mathcal{G}^*$ can be used for live-wire, minimum path segmentations.*

The smoothness energy term, E_s, is typically defined for segmentation to maximize the gradient between pixels of differing labels. For example, two common energies are:

$$E_s(p, l_p, q, l_q) = \begin{cases} 1/(\|\mathcal{I}_p - \mathcal{I}_q\|), or & (3) \\ e^{-\|\mathcal{I}_p - \mathcal{I}_q\|} & (4) \end{cases}$$

when $l_p \neq l_q$ and 0 otherwise. The smoothness energy is applied as costs to the edges of our graph (Fig. 2 (b)).

The data and smoothness energies are often defined independently to optimize the segmentation towards differing target criteria. Therefore for this work, we will decouple their visualization.

Live-wire Graph based segmentations like live-wire build a cut by the combination of partial segmentations. These partial segmentations are created by computing minimum paths on a dual graph $\mathcal{G}^*$. The edges of $\mathcal{G}^*$ are weighted with same smoothness cost on orthogonal edges in $\mathcal{G}$ (Fig. 2 (b)). Like the initial labeling for graph cuts, nodes are required to be denoted as constraints to produce a segmentation and are typically provided by a user or added semi-automatically based on a user's input. Given k user constraints given as points or partial boundaries, $C^* = \{c_1^*, c_2^*, \ldots c_k^*\}$ the minimization consists of finding the set of $k-1$ minimum paths, $\mathcal{P}^*$, where $\mathcal{P}^*{}_{ij}$ is a minimum path between consecutive constraints c_i^* and c_j^*. The optimization then minimizes the following energy:

$$E(L) = \sum_{\mathcal{P}_{ij}^* \in \mathcal{P}^*} \sum_{(p^*, q^*) \in \mathcal{P}_{ij}^*} dualE_s(p^*, q^*), \tag{5}$$

where $dualE_s(p^*, q^*)$ is the smoothness energy from $\mathcal{G}$ to which the edge (p^*, q^*) is orthogonal. If c_1^* equals c_k^*, then $\mathcal{P}^*$ gives a contour of a minimum cut of $\mathcal{G}$ under the user constraints. We can consider $E(L)$ in this case to only have a smoothness energy.

Finally, a live-wire approach can provide a segmentation of a minimal contiguous path of pixels in $\mathcal{G}$. In this case, the procedure is to produce a $k-1$ set of minimal paths, $\mathcal{P}$, given a set of user constraints, C, of size k minimizing some cost. For example,

$$E(L) = \sum_{\mathcal{P}_{ij} \in \mathcal{P}} \sum_{p \in \mathcal{P}_{ij}} \|\mathcal{I}_p - \mathcal{I}^{\dagger}\|, \tag{6}$$

produces a path minimally close to some target color, $\mathcal{I}^{\dagger}$. We can consider $E(L)$ in this case to only have a data energy.

4. Visualizing Segmentation Uncertainty

We now detail our new algorithms to compute a stability measure of graph-based segmentation for the purpose of uncertainty visualization. We first describe a fast and expressive visualization for live-wire segmentations based on their minimum path calculations. We then detail how the same approach can be extended to graph cuts segmentations and how it can further extend to a fast approximate solution. Finally, we describe how our approach enables the efficient query of a user-defined number of distinct alternative minimum cuts/paths. Pseudocode for the main algorithms are provided as supplemental material.

4.1. Live-wire Min-path Stability

In this sub-section, we introduce our stability measure for live-wire segmentations. Finding minimum pixel paths and partial segmentations for live-wire are the same operations on graphs, $\mathcal{G}$ and $\mathcal{G}^*$ respectively, and energies, Eq. 5 and Eq. 6 respectively. For simplicity, we will just describe our approach on pixel paths in $\mathcal{G}$. Computing a minimum path between each pair of consecutive user constraints, c_i and c_j, produces the minimal solution. Dynamic programming algorithms, like Dijkstra's algorithm [Dij59], can be run from one of the constraints in a pair to produce each minimal path in $O(|\mathcal{V}|\log(|\mathcal{V}|))$ time. A dynamic programming minimal path algorithm, in actuality, produces a full minimal path tree from a constraint to all nodes in the graph. We will refer to the minimum path tree for constraint c_i as $\mathcal{T}_i$. Finding the minimal path between any node in the graph and the constraint c_i is a simple traversal of the edges of $\mathcal{T}_i$ to the root, c_i. Let us define this traversal as $path(p, \mathcal{T}_i)$ for some node p. Note that the minimal path between constraints c_i and c_j is just $path(c_j, \mathcal{T}_i)$. If both trees, $\mathcal{T}_i$ and $\mathcal{T}_j$, are computed then we have already pre-computed all possible minimum paths that must pass through a new constraint, c_b, between c_i and c_j for all locations in our image [STP12]. This path is simply: $path(c_b, \mathcal{T}_i) \cup path(c_b, \mathcal{T}_j)$.

As part of the dynamic programming solution, there is also a cost recorded for each node in the tree that denotes the cost of the minimum path from that node to the constraint, $cost(p, \mathcal{T}_i)$. In this work, we visualize this cost as a stability field, noted $\mathcal{S}^{ij}$:

$$\begin{aligned} \mathcal{S}^{ij}(p) = \; & \left(cost(p, \mathcal{T}_i) + cost(p, \mathcal{T}_j)\right) \\ & - min_{\forall q \in \mathcal{G}} \left(cost(q, \mathcal{T}_i) + cost(q, \mathcal{T}_j)\right) \end{aligned} \tag{7}$$

where q is a point that minimizes the sum of the costs to c_i and c_j $\left(\mathcal{S}^{ij}(q) = 0\right)$. Intuitively $\mathcal{S}^{ij}$ encodes the cost of perturbing the minimal path through some node in our graph. In other words, $\mathcal{S}^{ij}$ encodes the stability or uncertainty in the optimal solution. Paths of low value (valleys) in $\mathcal{S}^{ij}$ represent alternative minimal paths that are possible from slight changes in the location of user input. Blur in the data would result in areas of thicker valleys in our field. Noise would thicken valleys as well, but also be chaotic as the minimum paths weave in and out of the noise. This distinction between smooth and chaotic gives a nice visual separation between uncertainty from the two different sources. Overall, for multiple constraints, our min-path stability measure is given by the field $\mathcal{S}$:

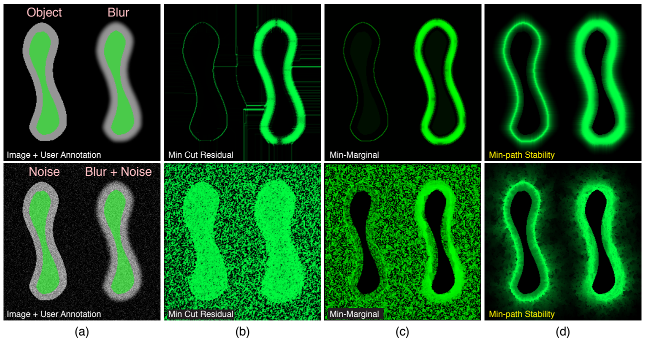

Figure 3: *Synthetic examples made from a grey object. (a) The rightmost copy has been blurred with a Gaussian filter. Gaussian noise is additionally added on the bottom. Initial foreground labeling is in green. (b) A naive stability measure is the residual of the minimum cut. Artifacts (erroneous ridges) can be misleading and the instability of the segmentation due to noise is obscured. (c) Visualizing the min-marginal energy [KT06] has similar problems. (d) Our visualization based on min-path stability encodes blur as thicker bands of uncertainty. Noise is also well represented as a chaotic "flowering" pattern in the bands.*

$$\mathcal{S}(p) = min_{\forall(i,j)} \left(\mathcal{S}^{ij}(p)\right) \tag{8}$$

The computation of $\mathcal{S}$ requires $O(|C||\mathcal{V}|\log(|\mathcal{V}|))$ steps, where C is the set of constraints in the input path. For visualization purposes, we invert $\mathcal{S}$, thus high values (ridges) encode higher uncertainty.

4.2. Graph Cuts Min-path Stability

In this sub-section, we extend our approach of stability evaluation for graph-cuts segmentations. For graph cuts, our approach first separates the min-path stability measure $\mathcal{S}$ for the data and smoothness energies as two separate fields $\mathcal{S}_d$ and $\mathcal{S}_s$ respectively. Of the two, $\mathcal{S}_d$ is the simplest to construct. Since data energy is often applied on a per pixel basis, $\mathcal{S}_d$ can just be an encoding of the data energy itself. For example, in our test cases we use the data energy of Li et al. [LSTS04] and visualize the foreground label energy. Therefore $\mathcal{S}_d = d_p^{\mathcal{F}} / (d_p^{\mathcal{F}} + d_p^{\mathcal{B}})$. We invert this field for the final visualization. Areas of high values in the visualization represent areas of the image that are likely to be labeled as foreground. Since this is a per node operation, the complexity of forming $\mathcal{S}_d$ is linear.

Regarding $\mathcal{S}_s$, a naive strategy would be to consider the residual of the minimum cut/maximum flow computation. Areas of uncertainty should coincide with edges in the graph that become saturated during the computation and therefore have low residual values. One can then construct $\mathcal{S}_s$ as the average residual about a node in the graph. We invert this field to allow high values to represent high uncertainty in the segmentation. Fig. 3 provides a synthetic

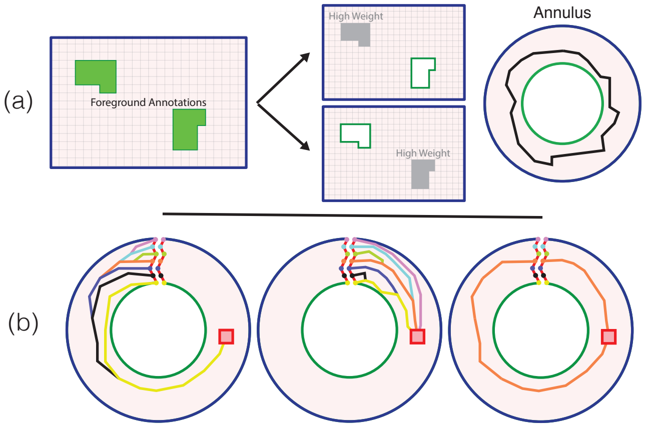

Figure 4: *Graph cuts uncertainty visualization. (a) Each connected component of the foreground annotation is considered a separate cut of our graph creating a logical annulus. (b) Each annulus is split by the minimum cost path from interior to exterior nodes. Given a node on the annulus, the minimum segmentation passing through it is constructed as the union of two minimum paths going to a split node. The related cost is the basis of our stability measure.*

example made from a grey object that has been corrupted with blur and gaussian noise. As shown in Fig. 3(b), this provides a reasonable but problematic approach due to artifacts (top: *notches* in the bands and ridges between objects and the image boundary). Noise in the data also overwhelms the visualization. Fig. 3(c) illustrates a visualization based on min-marginals [KT06,TA12]. This approach produces a good result for the top row, minus some notches in the blurred band. However, it is similar to the minimum cut residual in the presence of noise (bottom): the noise's effect on the segmentation is obscured by the overwhelming, global noise in the visualization. In contrast, our approach will present blur in the data as thickened bands and noise as chaotic, "flowering" bands in the vicinity of the segmentation boundaries. For the latter, this will only measure the local noise directly affecting the stability of the segmentation (Fig. 3(d)).

We will now describe our algorithm for the construction of $\mathcal{S}_s$ that leverages minimum path trees, similarly to Sec. 4.1. Given an initial labeling of foreground or background nodes, we consider each connected component of this labeling as a cut of our graph creating cyclic domains of our image (Fig. 4(a)). We will treat each connected component of the labeling as independent. We do this both to allow min-path stability to be formulated with a minimum path approach but also because this mimics the typical user behavior in segmentation. Users tend to label individual objects to be segmented with separate labels. Limitations of this independence are discussed in Section 6. We construct a set of logical rings, or *annuli*, from the cyclic domains formed by each foreground connected component cut, $F = \{f^1, f^2, \ldots\}$. Other foreground ($f^j, j \neq i$) and background labels are applied to each f^i as areas of high cost to enforce that segmentations do not pass through labeled areas (Fig. 4(a)).

We first construct min-path stability for each f^i using minimum path trees after cutting each annulus along a *split*, being the minimum cost path from the interior nodes of f^i to the exterior (Fig. 4(b)), as proposed by Itai and Shiloach [IS79]. The case where a foreground label touches the image boundary (collapsed annulus) is handled by logically extending the image by one pixel and labeling the extension as background. The nodes of the split, $n^i_j \in \mathcal{N}^i$, are replicated, $\hat{n}^i_j \in \hat{\mathcal{N}}^i$, across the split domain. We call all cuts that separate the interior of the annulus from the exterior as an *annulus segmentation*. Paths between each n^i_j and $\hat{n}^i_j$ define annulus segmentations and minimum paths between each n^i_j and $\hat{n}^i_j$ define a minimum annulus segmentation that must pass through n^i_j. The absolute minimum annulus segmentation is then the minimum of all these split node segmentations. As proposed by Summa et al. [SGSP15], we need not just compute paths from these nodes but full trees from both sides, $\mathcal{T}^i_j$ and $\hat{\mathcal{T}}^i_j$ for split node n^i_j. Like our live-wire example, each tree has both geometry and cost. Let us define $path(p, \mathcal{T}^i_j)$ and $cost(p, \mathcal{T}^i_j)$ to be the minimum path and cost from p to split node n^i_j using the minimum path tree $\mathcal{T}^i_j$. The cost of the minimum annulus segmentation for f^i that must pass through a node p is:

$$minCost(i,p) = \min_j(cost(p,\mathcal{T}^i_j) + cost(p,\hat{\mathcal{T}}^i_j)) \tag{9}$$

and the minimum annulus segmentation of f^i through p is:

$$minSeg(i,p) = path(p,\mathcal{T}^i_j) \cup path(p,\hat{\mathcal{T}}^i_j) \tag{10}$$

for the j used in $minCost(i,p)$ (Fig. 4(b)). We can record $minCost(i,p), \forall p$ during the construction of $\mathcal{T}^i_j$ and $\hat{\mathcal{T}}^i_j, \forall j$. We will note this minimum cost field as $\mathcal{S}^i_s$ for annulus f^i. To build a unified visualization for all annuli, we construct a new field $\mathcal{S}_s$:

$$\mathcal{S}_s(p) = min_i(\mathcal{S}^i_s(p)) \tag{11}$$

We invert this field for visualization to illustrate high uncertainty with high values. Finally, we enforce the property that segmentations cannot enclose background labelings by adding the minimum cost path from each connected component of the background labeling to the boundary as high cost in our uncertainty calculation.

This approach computes $|F|$ $|\mathcal{N}|$ minimum paths and therefore has complexity of $O(|F||\mathcal{N}||\mathcal{V}|\log(|\mathcal{V}|))$. $|F|$ is typically small enough to be considered a constant and $|\mathcal{N}|$ has a conservative upper bound of $\sqrt{|\mathcal{V}|}$. Therefore our time complexity can be considered to be: $O(|\mathcal{V}|^{1.5}\log(|\mathcal{V}|))$. This is lower complexity than the min-marginal procedure that require a brute-force application of $O(|\mathcal{V}|)$ graph cuts. Using the fastest algorithm for a graph cuts of a planar graph [STC09], min-marginal's complexity is $O(|\mathcal{V}|^2\log(|\mathcal{V}|))$.

Computational aspects If computed independently, the shortest path trees can have considerable redundancy as j increments in the computation of $\mathcal{T}^i_j$ or $\hat{\mathcal{T}}^i_j$. We report in the appendix (Sec. 7) a novel and time-efficient dynamic shortest path tree procedure that avoids redundant computation and adds practical speed-ups to our approach. Despite this accelerating procedure, the time complexity of this algorithm can still be limiting for usage in an interactive environment. Therefore, we introduce a fast, approximate solution that runs in $O(|\mathcal{V}|\log^2(|\mathcal{V}|))$ steps. Reif [Rei83] shows that the minimum path computations of Itai and Shiloach [IS79] in the split annulus cannot cross. Therefore, these paths can be used to divide the domain recursively, lending to a divide-and-conquer to re-

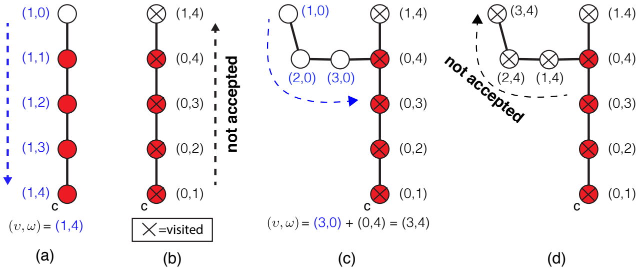

Figure 5: *Alternative Minimum Paths/Cuts procedure for a tree. (a) For each node in our queue, the path formed by a node is traversed, from its endpoint to the root, c, or a previously visited node. A running* down *count of* non-overlapping, υ, *(white) and* overlapping, ω, *(red) nodes to previously accepted paths are recorded. (b) If a path is not accepted, its* up *count is recorded and its nodes are marked as visited. (c) If the* down *traversal hits a node previously visited, the values used for acceptance testing (along with the values from the other tree) are the* down *count plus the recorded tuple of the intersecting node. (d) If the path is not accepted, walk down the path while adding the intersecting node's values. This process is continued until an acceptable path is found, i.e. if a non-rejected path with ratio of overlap to previous paths* $\alpha = \omega/(\omega+\upsilon)$ *is lower than a user threshold* β. *Then the bookkeeping is reset and re-initialized.*

cursively reduce the domain. See Summa et al. [SGSP15] for more details on the recursion. While not guaranteed to find the actual $minCost(i,p)$ for all $p \in \mathcal{G}$ for annulus f^i (see Sec. 5), this very efficient computation still finds the areas of high uncertainty present in $\mathcal{S}_s$ and provides a very good approximate solution.

4.3. Alternative Minimum Paths/Cuts

To this point, we have only been using the *cost* field of the minimum path trees. As we will show, we can also leverage the geometry (*path*) information to provide a user with a set of k distinct live-wire or per-labeling, graph cuts segmentations for the smoothness energy. This will allow a user to define and query a number of alternative segmentations, k, and a percentage of acceptable overlap between segmentations, β. In particular, a minimum path is considered to be *accepted* if its percentage α of nodes overlapping with the set AP of previously accepted paths is lower than β. As shown in Sec. 5, by a user adjusting β our fast extraction of AP is extremely useful to find k alternative segmentations.

Live-wire Given a minimum path between two user constraints c_i and c_j and their minimum path trees $\mathcal{T}_i$ and $\mathcal{T}_j$, the pair of constraints is associated with a min-path stability measure $\mathcal{S}^{ij}$ (Eq. 7). First, we create a priority queue Q of nodes in $\mathcal{G}$ prioritized ascending by $\mathcal{S}^{ij}$. Note that the front of Q will be filled with the nodes of the minimum path. At a high level, our algorithm will traverse every node p in the queue in ascending cost and will dynamically compute the corresponding α value, by traversing each of the two trees $\mathcal{T}_i$ and $\mathcal{T}_j$. If p has already been visited previously, we retrieve its previously computed α value. Otherwise, we traverse $path(p,\mathcal{T}_i)$ until it intersects a previously visited node. While traversing the path we also keep a temporary count of α as illustrated in Fig. 5(a) and (b). When an intersection occurs, the count of overlapping nodes is re-used from the previously computed paths, as illustrated in Fig. 5(c). If the test path is rejected given the user's β, we then traverse $path(p,T_i)$ from the intersection (or root) to p keeping another count, as illustrated in Fig. 5(d). We then pop a new node from our queue and continue testing. This continues until either the queue is empty or an acceptable path is found. If a path is accepted, it is added to the set AP and the process is restarted. Even though we need to reset our masks and bookkeeping to produce the correct solution, Q does not need to be reset since once a path defined by a node is rejected, it can be shown that it is rejected forever for an extraction. In our algorithm, for our bookkeeping we visit each node in the tree at most twice per path extraction. Therefore, the time complexity of the tree walking algorithm is $O(k|\mathcal{V}| + |\mathcal{V}|\log(|\mathcal{V}|))$. If $k < \log(|\mathcal{V}|)$ then the initial sort dominates the running time; otherwise it is linear per extracted path. k is typically very small since too many alternate paths can lead to a cluttered and confusing visualization.

Graph cuts Finding k distinct minimum annulus segmentations for the smoothness energy in graph cuts segmentations is a similar process. Since each path defined in $\mathcal{S}_s^i$ can come from one of a pair of trees from any of the split nodes, we must run the above process and keep the bookkeeping data for each $O(\sqrt{|\mathcal{V}|})$ number of trees. Also in this case, the accepted nodes will not be nicely contiguous as illustrated in Fig. 5 although the algorithm described will still compute the correct values. Therefore the algorithm is linear in the number of nodes per extraction, but only on each individual splitting path tree. The time complexity is then $O(k|F||\mathcal{N}||\mathcal{V}|)$. Since we give $|\mathcal{N}|$ an upper bound of $\sqrt{|\mathcal{V}|}$, assume that $|F|$ is small, and that k is very small for an interpretable visualization, the extraction has the same complexity as the min-path stability measure computation: $O(|\mathcal{V}|^{1.5})$.

5. Results

In this section, we provide examples of our new algorithms running on datasets from scientific and medical applications.

Live-wire Pixel Path Visualization. Fig. 6 illustrates a biomedical application with a confocal microscopy scan of the surface of a surgically removed prostate. For quality of life concerns, doctors and technicians would like to find and quantify the amount of nerve tissue removed during surgery. Nerves appear as faint lines of high luminance in the scan (Fig. 6(a)). With a minimum pixel path tracing, users can trace these nerves by setting the target color of Eq. 6 to be the maximum luminance value and adding constraints to the perceived start and end of the nerves. For the tracing on the right, an additional constraint is needed to account for noise in the data due to extraneous tissue. The minimum pixel paths are illustrated in red in Fig. 6 (b). In Fig. 6 (c), we show our uncertainty visualization for both tracings. First, the visualization illustrates that the tracings are located in slightly chaotic bands of high uncertainty due to the noise and blur in the data. Next, there are many ridges of high uncertainty therefore the constraints in the configuration will be highly sensitive to movement. In other words, it would not require much of a change in the constraint locations for an alternative, high ridge to become the new path tracing. Finally, the visualization has a very useful side effect of highlighting the nerve paths in the neighborhood around the trace. The visualization clearly outlines the faint nerves in the image. In Fig. 6 (d), using our alternative

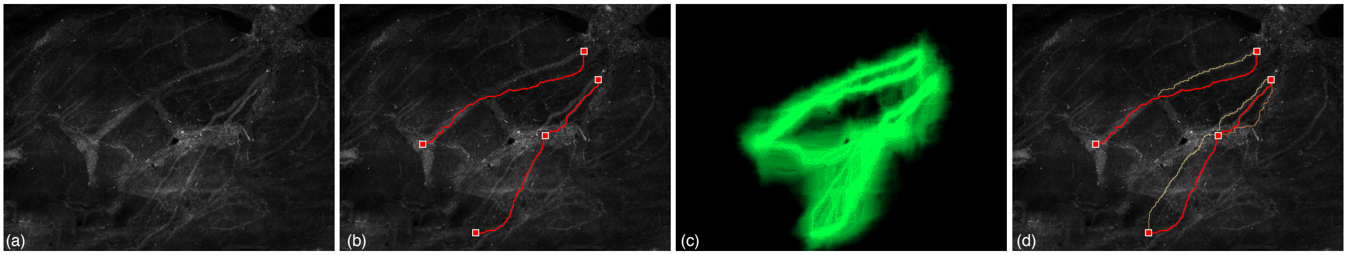

Figure 6: *Confocal microscopy image of a surgically removed prostate gland. (a) Nerve tissue appears as faint bright bands in the image although with much noise. (b) Using live-wire, a user can trace nerves as minimum paths of high luminance (red) between constraints (squares). (c) The min-path stability encodes where the paths could have gone. This not only encodes stability of the minimum paths and alternative paths, but also highlights possible nerves almost imperceptible in the source image. (d) With our approach a user can also extract alternative minimum paths (orange) to both quantify the stability of the trace and save effort by automatically finding other nerve paths (top path: $\beta = 0.5$, bottom path: $\beta = 0.2$). Image courtesy of J. Quincy Brown [Wea16].*

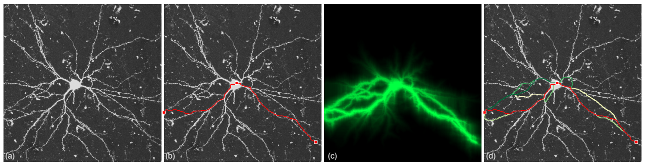

Figure 7: *(a) 2-photon microscopy image of a neuron from a macaque brain. (b) A user can trace two neural paths as minimum paths of high luminance (red) between constraints (squares). (c) The min-path stability illustrating the uncertainty of the tracing. Slight changes of the input can result in a variety of different paths for the data. (d) Our system allows users to find additional, distinct minimum paths (left path: $\beta = 0.7$; right path: $\beta = 0.8$) to both determine the stability of the trace, and also automatically extract alternate pathways. Image courtesy of Alessandra Angelucci.*

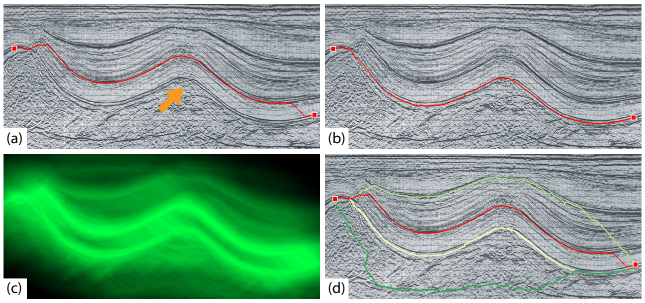

Figure 8: *2D time-migrated seismic section from the southern North Sea Silverpit Basin. (a) A horizon trace targeting a low luminance pixel path. The expected horizon (orange arrow) given the constraints is missed in favor of another (red). (b) Slight perturbation of the right constraint leads to a different trace. (c) The min-path stability illustrates the sensitivity of the constraints, leading to either horizon being a likely trace. (d) Our alternative extraction allows the identification of all horizons in this image ($\beta = 0.2$). Image courtesy of The Virtual Seismic Atlas user Simon Stewart.*

pixel paths query, the user can automatically find additional potential nerve paths.

Fig. 7 provides a neurobiology application of our technique with a 2-photon microscopy image of a neuron from a macaque brain (Fig. 7(a)). To gain deeper understanding of the brain, neural pathways are often traced. Very often this procedure is a manual or semi-automatic tracing of the paths in 2D. The tracing of the neural paths is a minimum pixel path tracing with the target color of Eq. 6 set to be the max luminance value. Constraints are set at the neuron and at the ends of the axons. The minimum path is illustrated in Fig. 7 (b) in red. In addition, in Fig. 7 (c) we provide the uncertainty visualization for our tracings. The many paths highlight the instability in the constraint positions and the thickness of the ridges encode the noise and blur in the data. Finally, all potential neural paths can be traced between the constraints with our alternative path extraction algorithm. These are highlighted in Fig. 7 (d) colored in green where saturation illustrates cost.

Fig. 8 provides a geoscience application of our technique visualizing the uncertainty in a horizon extraction in a seismic tomography image. Horizons are extracted by minimum pixel path tracing with the target color of Eq. 6 to be minimum luminance. In Fig. 8 (a), two constraints are added. Given their position, a user would likely expect to trace the horizon denoted by the orange arrow. The min-path stability in Fig. 8 (c) shows that the constraint position is unstable in that either horizon could be traced given a small perturbation of the constraints (as shown in Fig. 8 (b), where the right constraint has been slightly perturbed). Finally, Fig. 8 (d) illustrates how our alternative path extraction can automatically trace all 4 horizons in the image.

Graph Cuts Visualization. Fig. 9 shows an example of uncertainty in a graph cuts segmentation of a CT image of a liver with multiple metastases from pancreatic cancer (Fig. 9(a)). A user segments the image by labeling tumor areas as foreground (green) and liver as background (purple striped). The energy to be minimized (Eq. 1) is comprised of data energy from Eq. 2 and smoothness from Eq. 4. The graph cuts solution is indicated with transparent green (Fig. 9(b)). Unintuitively, tumors 1 and 2 are lumped into a single connected component in the segmentation. In addition, tumors 3 and 4 have an ambiguous separation and are almost joined. Visualizing the data uncertainty field in Fig. 9 (c), we can see that despite noise, the data energy seems to do an acceptable job of finding the tumors in the data. Our problem tumors blend in the data uncertainty but not significantly different than other tumors that are spatially close but segmented correctly. Looking at the min-path stability measure, we can find the problem (Fig. 9(d)). Tumors 1,2 and 3,4 are linked by very high ridges of uncertainty. Therefore their ambiguous separation is to be expected in the segmentation.

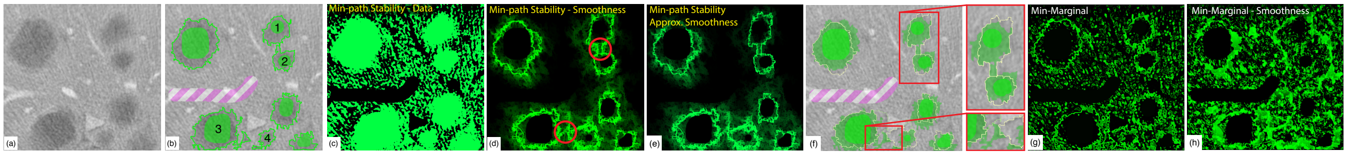
(a)
(b)
1
2
3
4
Min-path Stability - Data
(c)
Min-path Stability - Smoothness
(d)
Min-path Stability
Approx. Smoothness
(e)
(f)
Min-Marginal
(g)
Min-Marginal - Smoothness
(h)

Figure 9: *Tumor segmentation of a CT image of pancreatic cancer with multiple liver metastases. (a) The original image. (b) User annotation to denote tumor (green) and not tumor (purple striped). The graph cuts solution is provided in transparent green. Note that tumors 1 and 2 are considered a single region despite the user input and the separation between 3 and 4 is not clear cut. (c) The min-path stability for the data cost shows that the data energy isolates the tumors well but with much noise. More intensity of green denotes a higher likelihood that the pixel is considered tumor. (d) The min-path stability of the smoothness energy illustrates why there may be problems between 1-2 and 3-4. More intensity of green denotes a higher likelihood of the segmentation location given the user's labels. Notice the ridges of high intensity between the two problem boundaries. (e) Our approximate uncertainty visualization of the smoothness energy. Note that it still illustrates the major structures. (f) Our alternative segmentation calculation can be used to represent the uncertainty when overlaid on the segmentation* ($\beta = 0.8$)*. (g,h) Comparing to the visualization of the min-marginal energy [KT06] using both data and smoothness or just smoothness. While the pertinent structure exists, it is far more difficult to distinguish from the background noise in the data. Image courtesy of radRounds.com.*

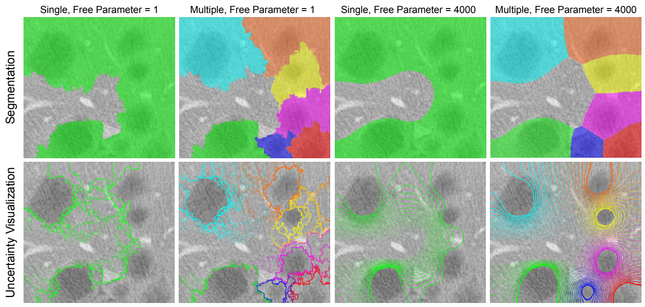
Single, Free Parameter = 1
Multiple, Free Parameter = 1
Single, Free Parameter = 4000
Multiple, Free Parameter = 4000
Segmentation
Uncertainty Visualization

Figure 10: *Random walker segmentation [Gra06] for our liver example (top). The segmentation produced, whether considering the foreground as a single or multiple labels or discounting the free parameter* (1) *or using the value from the previous work* (4000)*, bears little resemblance to the graph cuts segmentation of Fig. 9 (b). The uncertainty visualization of Praßni et al. [PRH10] (bottom) provides a good visualization of the random walker's stability, but cannot be generally applied to graph cuts.*

In Fig. 9 (e) we provide our fast min-path stability approximation, which captures the salient structure in the uncertainty. In Fig. 9 (f) our alternative annulus cuts yield segmentations ($k = 1$ for all tumors; $k = 2$ for tumors 3 and 4) that are distinct from the graph cut result. For tumors 1 and 2, their joining is stable even in the alternative segmentation and tumor 1 is primarily responsible for this merge. For tumors 3 and 4, the alternative segmentations in 4 intersect the minimum segmentation that encloses tumor 3, therefore indicating some instability in their separation. Finally, Fig. 9 (g) and 9 (h) show the min-marginal visualization of E and E_s respectively. Given the abundance of noise artifacts and general chaotic field of the visualization, the structure explaining our problem areas is difficult to distinguish and is therefore poorly represented.

Fig. 10 illustrates the significant differences between the random walker and graph cuts segmentations for our liver dataset and its effect on the visualization of Praßni et al. [PRH10]. Even when optimizing the same energy (Eq. 4 with free parameter = 1) and treating the foreground either as a single or multiple labels, both the segmentation and the visualization using the random walker approach bear little resemblance to the graph cuts segmentation. Moreover, Fig. 10 illustrates the significant smoothing effect the random walker's free parameter has on the segmentation. Here we show a value of 4000, as suggested in previous work [PRH10]. This parameter is generally uncommon in graph cuts energies. For these reasons, the visualization of Praßni et al. [PRH10] and our approach are not interchangeable. Random walker results are based on a reference implementation (scikit-image) of Grady's algorithm.

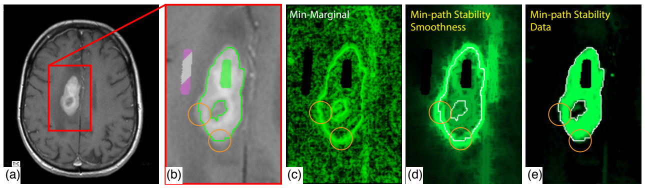

Figure 11: *Understanding the uncertainty of a segmentation algorithm by inspecting data and smoothness min-path stability. (a) A graph cuts segmentation of an MRI with contrast scan of CNS lymphoma in the left hemisphere of the brain. (b) User annotation denotes tumor (green) and not tumor (purple striped). The segmentation misses important structure (orange). (c) Min-marginal visualization [KT06] with a combined data and smoothness energies shows the uncertainty, although it is still unclear what is the cause. (d) Looking at the min-path stability measure there is uncertainty in this area that covers the areas missed by the segmentation. (e) Looking at the data term we can see it is ill defined to include this area and will need to be adjusted for future segmentations. Image courtesy of neurosciencecriticalcare.wordpress.com.*

Fig. 11 illustrates how the combination of our data and smoothness fields can be used to understand the behavior of a segmentation. In this example we have a graph cuts segmentation of an MRI with contrast scan of CNS lymphoma in the left hemisphere of the brain (Fig. 11(a)). User annotation denotes tumor (green) and not tumor (purple striped). The graph cuts solution is contoured in green. Highlighted in orange is important tumor mass missed by the segmentation (Fig. 11(b)). The data energy for the segmentation is Eq. 2 and smoothness is Eq. 4. The min-marginal visualization in Fig. 11 (c) shows the uncertainty in this area but gives no insight into the cause. If we look at the smoothness min-path stability field in Fig. 11 (d), some uncertainty in this boundary is due to the thickness of the ridge, but it still captures the extent of the tumor. The graph cuts solution is contoured in white. Fig. 11(e) provides further insight regarding the missing regions (orange circles). In par-

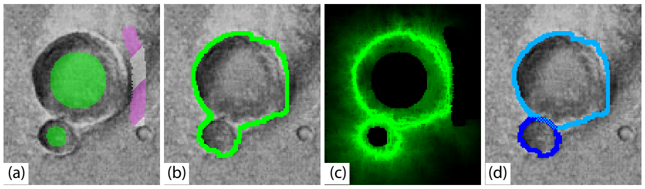

Figure 12: *Segmentation of craters in a Mars elevation map . (a) The interior of craters and their exterior are marked with foreground (green) and background (purple striped) constraints respectively. (b) The initial graph cuts segmentation identifies only one crater. (c) The smoothness min-path stability measure reveals the ambiguity of this segmentation result as two independent craters could have been identified. (d) Our approach enables users to recover these alternate segmentations. Image courtesy of NASA.*

ticular, it shows that this energy is ill defined to capture the problem areas and thus should be adjusted for the proper segmentation.

Fig. 12 presents an example from an elevation map of Mars. For such data-sets, the extraction of the geometry of the craters is important for the understanding of the properties of celestial bodies hitting the Martian soil, such as their age and origin. Here, the initial graph cuts segmentation identifies only one crater (Fig. 12(b)) despite the input, whereas the min-path stability clearly reveals the ambiguity of this segmentation as two independent craters could have been identified (Fig. 12(c)). Our approach easily enables users to recover these alternate segmentations (Fig. 12(d)).

Finally, Fig. 1 is an example that shows the generality of our technique. Fig. 1 shows another tumor segmentation from an MRI scan of the brain. In Fig. 1 (a) the graph cuts segmentation using the smoothness energy of Eq. 3, contoured in green, misses some significant tumor mass with user annotations to denote tumor (green) and not tumor (purple striped). Also in Fig. 1 (a), our alternative annulus segmentation can find an alternate path (blue) that encloses the missing tumor automatically. Fig. 1 (b) shows our smoothness min-path stability field. It illustrates that the noise and blur in the underlying data is depicted by chaotic banding and that the ridge denoting the alternate, correct segmentation can be seen (orange). We compare this result to our approximate smoothness min-path stability field in Fig. 1 (c), which shows that the fast approximation still captures the major structure of the uncertainty. We compare this visualization to the min-marginal energies in Fig. 1 (d) where the structure of the alternative segmentation is obscured and difficult to distinguish from the background noise in the data.

Live-wire Segmentation Visualization. Our work can also be applied to partial live-wire segmentations (Fig. 1(e)), where we segment the tumor using Eq. 3 and 5. Like the graph cuts example for this figure, we can use our alternative path extraction to find the segmentation (blue) that includes the tumor mass. Fig. 1 (f) shows the smoothness min-path stability field for the partial segmentation.

Performance We report and compare our performance in Table 1 on C++ implementations running on a system with an 2.8 GHz Intel Core i7. All runtimes are in seconds. First, we compare our running times against the min-marginal approach [KT06] (Tab. 1 (a) vs (e)). For the latter algorithm, we employed the graph cuts code by Kolmogorov and Boykov [BK04], which is widely considered a standard implementation and provides for graph reuse needed by the min-marginal approach to achieve practical running times, although alternatives could be considered [SK08].

	This Work				Min-Marginal	
	(a)	(b)	(c)	(d)	(e)	(f)
	Stand.	[SGSP14]	Update	Approx.	Stand.	Reuse
Fig. 1	4.4	0.2	0.3	0.04	78	0.2
Fig. 3a	9	0.6	0.6	0.1	174	0.1
Fig. 3b	7.3	0.8	0.5	0.2	240	2.4
Fig. 9	143.5	53	6.5	2.5	3432	8.7
Fig. 11	2.1	0.2	0.2	0.03	20.2	0.4

Table 1: *Performance results in seconds for our example datasets. For our technique, we provide timings for the standard calculation, using the calculation cut-off of the previous work [SGSP14], using our dynamic tree update routine, and our fast approximation. In addition, we provide the running times for the state-of-the-art min-marginal calculation [KT06] with and without tree reuse.*

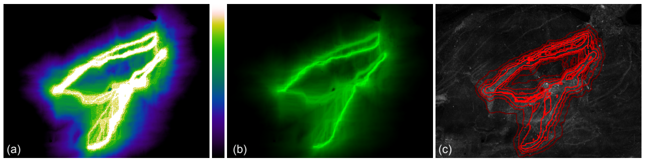

Figure 13: *Alternative approaches to visualize our stability field using (a) color maps motivated by perceptual studies [KRC02] (b) HDR gamma correction, and (c) isocontours of our stability field.*

Our lower time complexity algorithm is an order of magnitude faster. In addition, we compare our new dynamic shortest path tree update with previous work that uses a computation cut-off [SGSP14] (Tab. 1 (c) vs (b)). Our new procedure performs equivalently or up to an order faster than the previous work. We also compare our with dynamic update's performance with the min-marginal calculation with graph reuse [KT05] (Tab. 1 (c) vs (f)). Our technique computes a result with comparable or better time performance. Finally, Tab. 1 (d) shows that our approximate uncertainty visualization provides a very fast solution, even an order of magnitude faster than our dynamic update routine in some cases.

6. Discussion and Limitations

In this work, we presented a novel approach for the visualization of the uncertainty of graph-based 2D segmentation. Our approach is based on the formulation of a new stability measure, called min-path stability, which, for the first time, can be generally applied to two popular techniques, graph cuts and live-wire segmentations, with a single algorithm. As showcased with several real-life datasets, this measure highlights the ambiguous configurations where multiple, sometimes highly distinct, data segmentations are possible while being only separated by a small energy difference. In the presence of noise in the input data, this measure will reveal chaotic ridges, which will get progressively thicker in the presence of blur. Additionally, our measure will highlight many paths for sensitive input constraints. Beyond its adaptability, our algorithm has lower complexity than the current state-of-the-art for graph cuts only uncertainty, provides a new dynamic shortest path tree routine, gives a very faster approximate visualization, and allows a new query on the data to explore alternative segmentations efficiently. As future work, we feel similar approaches can be applied

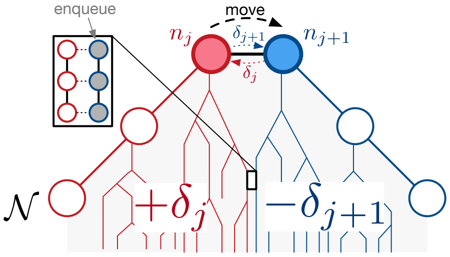

Figure 14: *Dynamic shortest path tree update from split path node n_j to n_{j+1} . Subtrees rooted at split path nodes $> j$ are decreased by the cost of edge (n_{j+1}, n_j). Subtrees rooted at split path nodes $\leq j$ are increased by the cost of edge (n_j, n_{j+1}). Dijkstra's priority queue is then fed with the decreased nodes on the front between these two forests.*

to other popular medical segmentation approaches such as active contours [HBG02,CBS*07,TOC10] and watershed cuts [SKK*12].

As mentioned in the algorithm description, assumptions that aid this work have limitations. For instance, we have found that the separation of the data and smoothness in our visualization is a good assumption given their typically independent formulations. For segmentations where they are intrinsically linked, the separation may not be clear, although we have not found examples where this is the case. Furthermore, an interesting direction for future work for k-segmentations when k > 2 is how to analyze the uncertainty yielding from solver inaccuracies or visualize the multi-dimensional instability . Treating of annuli as separate domains may introduce a bias into our algorithm. This bias gives greater weight to segmentations that extract individual labels over segmentations that join several, although the later will still be captured. Finding the absolute ordering without a complex combinatorial procedure is an interesting topic for future work. Our alternative segmentation currently provides results only on the smoothness cost. Extending this routine to factor in both data and smoothness costs with low complexity is an area of future work. In addition, if and how these alternate segmentations can be used intuitively in an interactive environment is an exciting area for future study. Finally, a natural future research direction is the extension of this approach to 3D data, which implies specific challenges in terms of computation, interaction, exploration and rendering.

Our stability field along with the fields produced by min-marginals and the random walker are real valued with often of high dynamic range. To apply more color resolution to areas with high instability, our system allows a user to threshold high field values before linear rescaling into byte range. This works well in most instances, but cases like Fig. 8 (c) can appear oversaturated. A change in colormap is insufficient to account for this large range. Fig. 13 (a) uses a color map motivated by perceptual studies [KRC02]. Another approach to reduce saturation is to apply a HDR compression technique. See Fig. 13 (b). These techniques by design will change relative magnitudes of areas of instability in the field and more study is required on how this data manipulation effects perceived structure. Finally, it is also possible to overlay the contours of the stability field similar to Praßni et al. [PRH10]. See Fig. 13 (c). While promising, we have found that there are still open questions for their best use. For example, the ideal number of isocontours is data dependent and has competing requirements in that too little contours will miss important structure and too many will obscure the underlying image. How best to visualize these fields, especially in a way that is intuitive for domain users, is an interesting area of future work.

7. Appendix: Dynamic Shortest Path Tree Update

Given a shortest path tree, $\mathcal{T}^i_j$, rooted at split node n^i_j, the tree can be reused in computation of $\mathcal{T}^i_{j+1}$ rooted at n^i_{j+1} (Fig. 14). First, consider the minimum split path $\mathcal{N}$ (Fig. 4(b)). This path also exists in $\mathcal{T}^i_j$. We can then consider the tree $\mathcal{T}^i_j$ as a forest where each subtree is rooted at a split path node and each root is connected by our split path. This structure can be recorded during the initial $\mathcal{T}^i_j$ computation. Consider the forest associated with split nodes that are greater than j. By moving the root to n_{j+1} all nodes of these subtrees will be reduced by the energy cost, δ_{j+1}, of edge $(n_{j+1}, n_j) \in \mathcal{E}$. Similarly, subtrees rooted at split path nodes $\leq j$ are increased by the energy cost, δ_j, of edge (n_j, n_{j+1}). Increasing or decreasing the costs in a shortest path tree yields another valid shortest path tree. Therefore the subtrees that are decreased exist in $\mathcal{T}^i_{j+1}$ with this reduced cost and can be skipped by the new tree calculation. Moreover, a portion of the increased subtrees will not change, save the portions affected by the subtrees that are decreased. We can use these properties for an efficient computation. First, all subtrees with a root $\leq j$ are increased. Next, all subtrees with a root $> j$ are decreased. Finally, if using Dijkstra's algorithm, our priority queue is then fed with the decreased nodes on the front between these two forests. This leads to a very efficient update. In our testing, we have seen this update on a random energy to be bound by the length of the front, $O(\sqrt{|\mathcal{V}|})$. On the random field, the compute cost was reduced by two orders of magnitude over independent computation.

Acknowledgments

This work is partially supported by NSF:CRII 1657020, NSF:CGV 1314896, NSF:IIP 1602127 NSF:ACI 1649923, DOE/SciDAC DESC0007446, CCMSC DENA0002375, PIPER: ER26142 DESC0010498, and Bpifrance grant "AVIDO" (Programme d'Investissements d'Avenir P112017-2661376/DOS0021427).

References

[ATHL14a] AL-TAIE A., HAHN H., LINSEN L.: Uncertainty estimation and visualization in proababilistic segmentation. *Comp. & Graph.* (2014). 2

[ATHL14b] AL-TAIE A., HAHN K., LINSEN L.: Uncertainty-aware ensemble of classifiers for segmenting brain mri data. In *VCBM* (2014). 2

[BDVJ*16] BROWET A., DE VLEESCHOUWER C., JACQUES L., MATHIAH N., SAYKALI B., MIGEOTTE I.: Cell segmentation with random ferns and graph-cuts. *arXiv preprint arXiv:1602.05439* (2016). 2

[BHJ*14] BONNEAU G.-P., HEGE H.-C., JOHNSON C. R., OLIVEIRA M. M., POTTER K., RHEINGANS P., SCHULTZ T.: Overview and state-of-the-art of uncertainty visualization. In *Scientific Visualization*. Springer, 2014, pp. 3–27. 2

[BK04] BOYKOV Y., KOLMOGOROV V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI 26*, 9 (2004), 1124–1137. 2, 9

[BVZ01] Boykov Y., Veksler O., Zabih R.: Fast approximate energy minimization via graph cuts. *PAMI 23*, 11 (2001), 1222–1239. 2

[BYGRS12] Batra D., Yadollahpour P., Guzman-Rivera A., Shakhnarovich G.: Diverse m-best solutions in markov random fields. In *ECCV* (2012), Springer, pp. 1–16. 2

[CBS*07] Cobzas D., Birkbeck N., Schmidt M., Jagersand M., Murtha A.: 3d variational brain tumor segmentation using a high dimensional feature set. In *ICCV* (2007), IEEE, pp. 1–8. 10

[CCP07] Collins C., Carpendale S., Penn G.: Visualization of Uncertainty in Lattices to Support Decision-Making. In *Symp. on Vis.* (2007). 2

[Del91] Dellepiane S.: Image segmentation: errors, sensitivity, and uncertainty. In *EMBS* (1991). 2

[Dij59] Dijkstra E. W.: A note on two problems in connexion with graphs. *Numerische Mathematik 1* (1959), 269–271. 4

[DLD*15] Dai S., Lu K., Dong J., Zhang Y., Chen Y.: A novel approach of lung segmentation on chest ct images using graph cuts. *Neurocomputing 168* (2015), 799–807. 2

[Eea13] Emblem K. E., et al.: Vessel architectural imaging identifies cancer patient responders to anti-angiogenic therapy. *Nature medicine 19*, 9 (2013), 1178–1183. 1

[FLBF*06] Funka-Lea G., Boykov Y., Florin C., Jolly M.-P., Moreau-Gobard R., Ramaraj R., Rinck D.: Automatic heart isolation for ct coronary visualization using graph-cuts. In *ISBI* (2006), IEEE, pp. 614–617. 2

[Gra06] Grady L.: Random walks for image segmentation. *PAMI 28*, 11 (2006), 1768–1783. 2, 8

[HBG02] Ho S., Bullitt E., Gerig G.: Level-set evolution with region competition: automatic 3-d segmentation of brain tumors. In *ICPR* (2002), vol. 1, IEEE, pp. 532–535. 10

[HBG*11] Höllt T., Beyer J., Gschwantner F., Muigg P., Doleisch H., Heinemann G., Hadwiger M.: Interactive seismic interpretation with piecewise global energy minimization. In *PacificVis* (2011), IEEE, pp. 59–66. 2

[HBS*12] Heckel F., Braunewell S., Soza G., Titjen C., Hahn H.: Sketch-based image-independent editing of 3d tumour segmentation using variational interpolation. In *VCBM* (2012). 2

[HMTH13] Heckel F., Moltz J., Titjen C., Hahn H.: Sketch-based editing tools for tumour segmentation in 3d medical images. *CGF* (2013). 2

[HPD95] Hannah I., Patel D., Davies R.: The use of variance and entropic thresholding methods for image segmentation. *PR* (1995). 2

[IS79] Itai A., Shiloach Y.: Maximum flow in planar networks. *SIAM Journal on Computing 8*, 2 (May 1979), 135–150. 5

[JM10] Jagadeesh V., Manjunath B.: Interactive graph cut segmentation of touching neuronal structures from electron micrographs. In *ICIP* (2010), IEEE, pp. 3625–3628. 2

[KMW*15] Kuo J.-w., Mamou J., Wang Y., Saegusa-Beecroft E., Machi J., Feleppa E. J.: A novel nested graph cuts method for segmenting human lymph nodes in 3d high frequency ultrasound images. In *ISBI* (2015), IEEE, pp. 372–375. 2

[KRC02] Kindlmann G., Reinhard E., Creem S.: Face-based luminance matching for perceptual colormap generation. In *VIS* (2002). 9, 10

[KT05] Kohli P., Torr P. H.: Efficiently solving dynamic markov random fields using graph cuts. In *ICCV* (2005), vol. 2, pp. 922–929. 9

[KT06] Kohli P., Torr P. H. S.: Measuring uncertainty in graph cut solutions: Efficiently computing min-marginal energies using dynamic graph cuts. In *ECCV* (2006), pp. II: 30–43. 1, 2, 4, 5, 8, 9

[LRCP07] Lee B., Robertson G. G., Czerwinski M., Parr C. S.: Candidtree: visualizing structural uncertainty in similar hierarchies. *Information Visualization 6*, 3 (2007), 233–246. 2

[LSTS04] Li Y., Sun J., Tang C.-K., Shum H.-Y.: Lazy snapping. *TOG 23*, 3 (2004), 303–308. 3, 4

[MB98] Mortensen E. N., Barrett W. A.: Interactive segmentation with intelligent scissors. *Graph. Mod. & Im. Proc.* (1998). 2

[PGA13] Potter K., Gerber S., Anderson E.: Visualization of uncertainty without a mean. *CGA* (2013). 2

[PRH10] Prassni J.-S., Ropinski T., Hinrichs K.: Uncertainty-aware guided volume segmentation. *TVCG* (2010). 2, 3, 8, 10

[PWH11] Pöthkow K., Weber B., Hege H.-C.: Probabilistic marching cubes. In *CGF* (2011), vol. 30, pp. 931–940. 2

[Rei83] Reif J. H.: Minimum $s-t$ cut of a planar undirected network in $O(n\log^2(n))$ time. *SIAM J. Comput. 12*, 1 (Feb. 1983), 71–81. 5

[SG07] Sinop A. K., Grady L.: A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In *ICCV* (2007). 2

[SGSP14] Summa B., Gooch A., Scorzelli G., Pascucci V.: *Towards Paint and Click: Unified Interactions for Image Boundaries*. Tech. Rep. UUSCI-2014-004, University of Utah, 2014. 9

[SGSP15] Summa B., Gooch A., Scorzelli G., Pascucci V.: Paint and Click: Unified Interactions for Image Boundaries. *CGF* (2015). 5, 6

[SHM10] Saad A., Hamarneh G., Moller T.: Exploration and visualization of segmentation uncertainty using shape and appearance prior information. *TVCG 16*, 6 (2010), 1366–1375. 2

[SK08] Schraudolph N., Kamenetsky D.: Efficient exact inference in planar ising models. In *NIPS* (2008). 9

[SK13] Shultz T., Kindlmann G.: Open-box spectral clustering: Applications to medical image analysis. *TVCG* (2013). 3

[SKK*12] Straehle C.-N., Koethe U., Knott G., Briggman K., Denk W., Hamprecht F. A.: Seeded watershed cut uncertainty estimators for guided interactive segmentation. In *CVPR* (2012). 10

[SMH10] Saad A., Möller T., Hamarneh G.: Probexplorer: Uncertainty-guided exploration and editing of probabilistic medical image segmentation. *CGF* (2010). 2

[SS04] Sezgin M., Sankur B.: Survey over image thresholding techniques and quantitative performance evaluation. *Elec. Imag.* (2004). 2

[STC09] Schmidt F. R., Toppe E., Cremers D.: Efficient planar graph cuts with applications in computer vision. In *CVPR* (2009). 5

[STP12] Summa B., Tierny J., Pascucci V.: Panorama weaving: fast and flexible seam processing. *TOG 31*, 4 (July 2012), 83:1–83:11. 4

[TA12] Tarlow D., Adams R. P.: Revisiting uncertainty in graph cut solutions. In *CVPR* (2012), IEEE, pp. 2440–2447. 2, 5

[THA10] Top A., Hamarneh G., Abugharbieh R.: Spotlight: Automated confidence-based user guidance for increasing efficiency in interactive 3d image segmentation. In *MICCAI MCV* (2010). 2

[TOC10] Taheri S., Ong S. H., Chong V.: Level-set segmentation of brain tumors using a threshold-based speed function. *IVC* (2010). 10

[Wea16] Wang M., et al.: Gigapixel surface imaging of radical prostatectomy specimens for comprehensive detection of cancer-positive surgical margins using structured illumination microscopy. *Scientific Reports 6*, 27419 (2016). 7

[WMK13] Whitaker R. T., Mirzargar M., Kirby R. M.: Contour boxplots: A method for characterizing uncertainty in feature sets from simulation ensembles. *TVCG 19*, 12 (2013), 2713–2722. 2

[WZS*13] Waggoner J., Zhou Y., Simmons J., Salem A., De Graef M., Wang S.: Interactive grain image segmentation using graph cut algorithms. In *IS&T/SPIE Electronic Imaging* (2013). 2

[WZW04] Warfield S. K., Zou K. H., Wells W. M.: Simultaneous truth and performance level estimation (staple): an algorithm for the validation of image segmentation. *T-MI 23*, 7 (2004), 903–921. 2

[ZBS01] Zhang Y., Brady M., Smith S.: Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. *T-MI 20*, 1 (2001), 45–57. 2