Seeded watershed cut uncertainty estimators for guided interactive segmentation

Christoph-N. Straehle* Ullrich Koethe * Graham Knott†
Kevin Briggman‡ Winfried Denk§ Fred A. Hamprecht*

July 16, 2012

Abstract

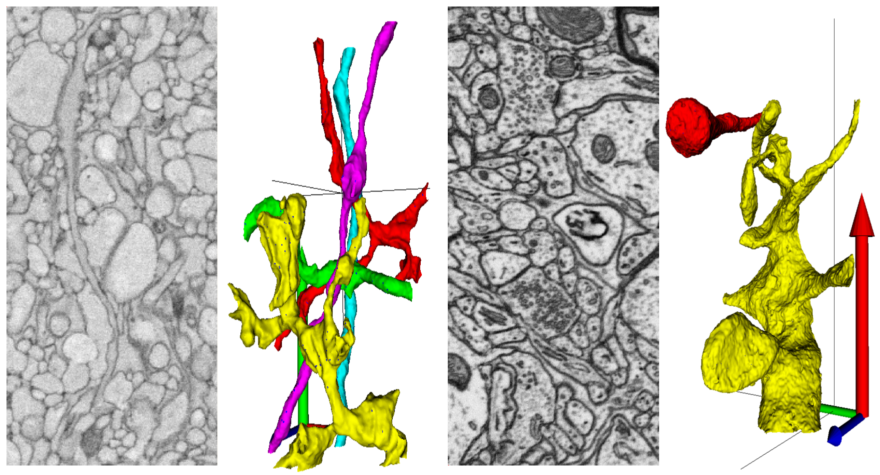
Watershed cuts are among the fastest segmentation algorithms and therefore well suited for interactive segmentation of very large 3D data sets. To minimize the number of user interactions ("seeds") required until the result is correct, we want the computer to actively query the human for input at the most critical locations, in analogy to active learning. These locations are found by means of suitable uncertainty measures. We propose various such measures for watershed cuts along with a theoretical analysis of some of their properties. Extensive evaluation on two types of 3D electron microscopic volumes of neural tissue shows that measures which estimate the non-local consequences of new user inputs achieve performance close to an oracle endowed with complete knowledge of the ground truth.

1 Introduction

Interactive segmentation is a popular paradigm in image analysis because it combines the number-crunching capabilities of a computer with the high-level understanding of a human. When the segmentation result is immediately updated after each interaction, the user can readily spot errors and correct these by a (hopefully small) number of additional inputs. Unfortunately, this elegant scheme breaks down in 3D because errors no longer "pop out" to the user's attention as they do in 2D – it is not possible to visualize complicated 3D segmentations in a way that makes user inspection and intervention as easy as in two dimensions. Most commonly, volume data is presented on a 2D screen by means of three orthogonal views, and the user has to scroll through several, and possibly many, layers in order to find or rule out segmentation errors.

We propose to solve this problem by guided interactive segmentation [1], akin to active learning. Active learning (AL) [2] schemes aim at the steepest possible learning

*HCI University of Heidelberg, 69117 Heidelberg
†EPF, Lausanne, CH-1015 EPFL SV-GE
‡National Institute of Health, Bethesda, MD 20892
§MPI for Medical Research, 69120 Heidelberg



(a) (b)

Figure 1: Raw data and ground truth. (a) Serial blockface electron microscopy (SBEM) image slice and 3D rendering of some neural processes from the image stack. (b) Focused ion beam electron microscopy (FIBSEM) image slice and two neural processes.

curve by querying for user input on locations which are regarded as most informative by a suitable selection criterion, so that user effort is focused on decisions with high impact. Accordingly, our algorithm not only proposes a segmentation based on the user's inputs, but also estimates a *confidence* in the segmentation result which will guide the user to locations where the uncertainty is highest. Good uncertainty criteria are especially challenging in our context because segmentation quality is a non-local property: A very small error (e.g. a single wrongly deleted boundary) can have catastrophic global consequences (such as an erroneous merge of two very large regions). Purely local error estimates as used in most existing AL work on interactive segmentation [3, 1, 4] are not sufficiently sensitive to these non-local effects.

Our interactive segmentation framework is based on the watershed algorithm because it has a small computational footprint and is attractive for our target application: Microscopic images of neural tissue (see Fig. 1) are composed of very thin and elongated structures, which may pose a problem for the graph cut and random walk algorithms with their well-known shrinking bias.

For the neurobiological application example we compute watershed cuts on supervoxel graphs [5, 6] (see section 4.2). The goal of interactive segmentation is therefore to merge all supervoxels belonging to the same object. User labels are interpreted as seeds for either a foreground object or the background, and regions are defined according to a watershed cut initiated by these seeds [7]. Our AL criteria relate to the common boundaries between adjacent supervoxels. They take into account the relevance of the boundary with regard to the uncertainty in other regions.

Specifically we make the following contributions:

- We define and characterize a number of uncertainty criteria that can be used in the context of interactive segmentation with the watershed cut algorithm.
- We conduct extensive comparisons of the practical performance of these criteria in 3D neuro-imaging application examples.
- We demonstrate empirically that correct segmentations are achieved much faster when user attention is guided by our best active learning criteria.

2 Background and related work

Interactive segmentation algorithms must be able to take user input ("seeds") into account and update results incrementally when new input arrives, and it must be sufficiently fast to ensure interactive response times. Some of the most important seeded segmentation methods can be unified in terms of the *power-watershed* framework [8]. It defines a segmentation as a labeling of a graph $G(V, E)$, where the optimal labeling $\mathbf{x}$ minimizes an energy function

$$\begin{aligned} E(\mathbf{x}, \mathbf{w}) = \sum_{v_i \in V} w_{0,i}^p \|x_i\|^q + w_{1,i}^p \|1 - x_i\|^q \\ + \sum_{e_{ij} \in E} w_{ij}^p \|x_i - x_j\|^q \end{aligned} \tag{1}$$

where $x_i \in L$ is the label associated with node $v_i \in V$, $\mathbf{x}$ is the vector of all label assignments, and w_i, w_{ij} are node and edge weights, respectively. The first sum thus measures compatibility of the labeling with a given region model, whereas the second sum enforces smoothness of the solution. For different exponents, the power watershed specializes to the watershed cut algorithm ($p \to \infty$, q finite, [9], [7]), the random walker (p finite, $q = 2$, [10]) and an Ising-type Markov random field amenable to graph-cut segmentation (p finite, $q = 1$, [11]). A number of coarse-grained [12], precomputed [13] or warm-started [14] strategies have been suggested to speed up the response to user input.

A significant simplification of the solution space is achieved by moving from a grid-graph defined on the original voxels to a coarser graph of supervoxels. The weighted graphs that reflect the supervoxel adjacency typically have a large total number of nodes, a small number of seeded nodes (the user scribbles), and are sparse (i.e. the number of edges is of the same order as the number of nodes). This is a favorable situation for power-watershed methods.

A number of user guidance schemes have already been proposed in the context of interactive segmentation for the random walker [1, 4] or graph cuts [3, 15]. These works use uncertainty cues based on the margin in the case of the random walk or min-marginal energies in the graph cut case [16], both of which capture mostly local information. In addition a perturbation based local uncertainty estimator for the graph-cut has recently been proposed in [15].

We propose several non-local uncertainty estimators for the watershed cut, and show that they perform better than a local alternative. Closest in spirit to our work are the stochastic topological watershed variants that have been proposed for *un*seeded segmentation [17, 18]. These authors consider a topological watershed from randomized seeds, while we randomize the edge weights [7] instead.

3 Uncertainty measures for watershed cuts

After reviewing the notion of a watershed cut and its relation to a minimum spanning tree (MST) in Section 3.1 we will present two different types of uncertainty estimators. The first type presented in Section 3.2 is based on a minimal perturbation property and expresses how the obtained segmentation boundary depends on single edges in a graph. The second type of uncertainty estimators in Section 3.3 takes into account the uncertainty of the edge weights themselves. It measures how much the overall segmentation changes when sampling noisy edge weights.

3.1 Watershed cuts and minimum spanning trees

The interactive segmentation algorithm in this paper is based on watershed cuts [9, 7]. It starts from a supervoxel graph which is computed by a standard flooding-type watershed algorithm [19] on a suitable boundary indicator, in our case the largest eigenvalue of the Hessian matrix which measures "ridgeness" and thus indicates cell membranes. The region adjacency graph $G(V, E)$ of the supervoxels is equipped with edge weights that encode surface strength (in particular, the minimum value of the boundary indicator on the corresponding surface patch). User seeds provide hard assignments of some supervoxels to the background or one of the foreground regions.

It is well known [7] that the watershed cut is equivalent to a minimum spanning tree (MST) computation on a suitably augmented graph $G'(V', E')$ that contains a supernode v_0 connected to seed nodes $v_{-l}, l \in L$ for each class type that are connected to v_0 with zero weight edges. All labeled nodes (i.e. all supervoxels holding a user seed) are also connected to these seed nodes with zero-weight edges, which are guaranteed to remain in the MST. Once the MST with root node v_0 has been constructed, subtrees originating from seed nodes v_{-l} form segments of the final segmentation. The subtrees and the segmentation of the seeded watershed cut are defined as follows:

Definition 3.1 (Subtree T^i) *Let T be a spanning tree of G with root node v_0. The subtree $T^i = (V^i, E^i)$ is defined as the set of nodes V^i and edges E^i which in T can only be reached from the root node by a path containing v_i.*

Definition 3.2 (Segmentation $\mathbf{x}$) *Let T be a spanning tree of G with root node v_0. The label assignment of all nodes is called the segmentation $\mathbf{x}$ with $x_i = l$ if node v_i is element of the subtree T^{-l} of seed node v_{-l}. Thus, all nodes i with label $x_i = l$ and the edges connecting them form the subtree T^{-l} of T with root node v_{-l}.*

Since we are concerned with non-local uncertainty estimates that measure influences on the segmentation, we frequently rely on the definition of the edges connecting different segments:

Definition 3.3 (Cut set $C(T)$) *Let T be a spanning tree of G. An edge $e = (i, j)$ is element of the cut set $C(T)$ if the vertices v_i and v_j belong to different subtrees T^{-l} so that $x_i \neq x_j$.*

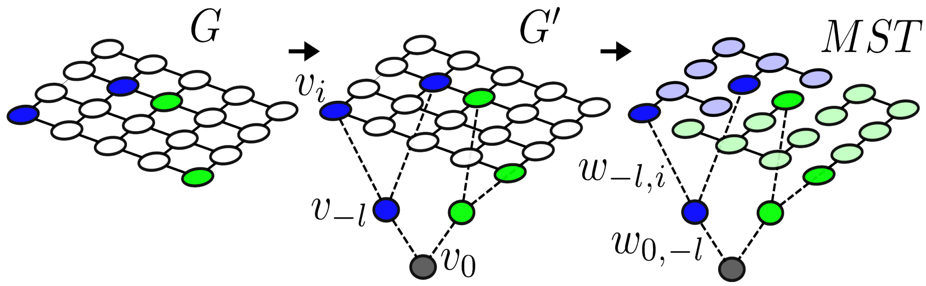


Figure 2: Illustration of the graph construction process. Colored nodes correspond to user seeds. Shown are the original graph G, the modified graph G' and a minimum spanning tree (MST) with the resulting segmentation label assignment indicated by light colors.

In the following, we will introduce estimators arising from two different general principles. The first one analyzes the effect of single edge weight perturbations on the MST and the resulting segmentation, in a manner similar to [20]. The second estimator takes into account that the edge weights of the supervoxel graph are themselves subject to uncertainty, and measures how much the segmentation would change under perturbations of all weights.

As a baseline we compare our non-local estimators to a *local instability* defined by the margin of the seeeded watershed cut: It is computed from the maximum weight $w_{kl}, (k,l) \in P(i,0)$ that appears in the path $P(i,0)$ between a node i and the root node 0 in the MST, when only considering seeds of one type. The margin is the difference between the lowest possible path from a seed type and the next best one from another seed type.

3.2 Link instability via minimum perturbations

The first structural uncertainty measure we propose estimates the influence of *individual* edges on the final segmentation. In Lemma 7.3 (Appendix) we show that only the inclusion of edges $f \in C(T)$ from the current cut set $C(T)$ (i.e. $f \notin T$) into the minimum spanning tree changes the resulting segmentation. We propose to measure the instability of all edges $e \in T$ currently part of the MST T by calculating how often an edge e would be removed from the MST when considering minimal perturbations that would enforce the inclusion of an edge $f \in C(T), \notin T$ into the MST. i.e. minimal perturbations that change the segmentation. Thus, our uncertainty estimator measures how much the overall segmentation depends on $w(e)$ being being smaller than $w(f)$ considering a minimum perturbation of $w(f)$ that would change the segmentation.

The minimal perturbation neccessary to enforce inclusion of a cut edge f into the MST is characterized by the following Definition:

Definition 3.4 (Minimal edge weight perturbation) *The minimal edge weight perturbation $\hat{w}_{ij}$ of some edge $f = (i,j) \notin T$ that enforces the inclusion of this edge in a minimum spanning tree of G is given by $\hat{w}_{ij} = \min_{(k,l)} w_{ij} - w_{kl} + \epsilon, (k,l) = e \in T$*

This definition follows from Lemma 7.2 in the Appendix: a smaller perturbation does not lead to any negative e, f exchange.

Algorithm 3.2.1 counts how often an edge which is part of the MST would be removed from the spanning tree considering all minimal edge weight perturbations (Definition 3.4) that change the segmentation (Definition 3.2) induced by the MST. The correctness of Algorithm 3.2.1 is proven in the Appendix (Lemma 7.4).

Algorithm 3.2.1 Link Instability

---

1. Determine the cut set $C(MST)$.
2. Do a breadth first search starting from the root node of the MST and store at each node the edge with maximum weight encountered so far.
3. For each edge in the cut set: The minimal exchange partner is the edge with maximum weight stored in either end node of the cut edge. Increment the counter of that edge.

We note that the runtime of Algorithm 3.2.1 is linear in the number of edges of the graph and thus preserves the low computational overhead of the watershed cut. This follows from the fact that the computational complexity of the determination of $C(T)$, the breadth first search and the counter incrementation are all linear in the number of edges. We propose to use the *link instability* as a uncertainty measure for the edges.

3.3 Uncertainty from stochastic graphs

Edge weights in the supervoxel graph are computed from local features. Since the raw data are noisy, the edge weights are necessarily noisy as well. We accomodate this uncertainty by moving from deterministic edge weights to stochastic ones, which are distributed according to a probability distribution reflecting the noise. In contrast to [17, 18] who obtain a stochastic watershed by random perturbations of the seed positions, we keep the seeds fixed and instead randomize the edge weights. In particular, we define the stochastic watershed cut by replacing the original edge weights w_{ij} with $w'_{ij} \sim PD_{ij}$, where PD_{ij} is the weight distribution of edge (i, j) that can be modeled by multiplicative noise, for example.

Consequently, the hard label assignment x_i of node v_i in the fixed-weight watershed cut will be replaced by the probability of a label assignment $p_i(l), l \in L$ which depends on the edge weight distribution PD_{ij} of *all* edges (i, j).

Ideally, we would like to compute these probabilities exactly, but the analysis in the Appendix and Supplementary Material shows that no efficient algorithm for this problem exists. This is why we study an approximation.

3.3.1 Sampling scheme

Since computing the exact label distribution is infeasible, we propose to sample t_{max} complete graphs $G^t, t \in \{1, .., t_{\text{max}}\}$ from the space of feasible graphs by sampling their edge weights w^t_{ij} from the independent probability distributions PD_{ij}.

For each randomly drawn graph G^t, the seeded watershed cut is computed by calculating the minimum spanning tree and assigning the node labels $x_i^t = l$ according to Definition 3.2. After all repetitions, the probability of node v_i carrying label l can be estimated as $p_i(l) = \frac{1}{t_{\max}} \sum_{t'=1}^{t_{\max}} \delta(x_i^{t'}, l)$. The final segmentation after $t_{\max}$ repetitions is defined as $x_i = \underset{l}{\operatorname{argmax}}\, p_i(l)$, i.e. x_i is assigned in a winner-take-all fashion to the label l to which node i was most often assigned during the trials.

The computational complexity of this sampling scheme depends linearly on the number of sampled graphs and the individual minimum spanning tree computations can be executed in parallel.

3.3.2 Stochastic uncertainty estimators

Uncertainty estimators based on the stochastic watershed cut can be defined in various ways, a natural one being the probability margin between the winning label $x_i = l$ and the one with the next highest class count, i.e. $m_i = p_i(l) - p_i(z')$ where $z' = \underset{\mathbf{l'\neq l}}{\operatorname{argmax}}\, c_i^{l'}$.

However, this is only a local estimator of uncertainty, whereas critical edges should be characterized by their *non-local* effects. By combining the link instability according to Algorithm 3.2.1 with stochastic watershed cuts by accumulating the link instability of the edges over all $t_{\max}$ trials, a measure incorporating the influence of a single edge on the *global* segmentation can be obtained. This estimator is called *stochastic link instability*.

Algorithm 3.3.1 Stochastic segmentation instability

- Do $t_{\max}$ times
  1. Construct graph G^t by sampling w_{ij}^t from PD_{ij}.
  2. Construct $MST(G^t)$ with root node 0, and store the segmentation x_i^t.
  3. Calculate and store the size of the subtree $h_i^t = |T^i|$ of each node i.
- Calculate $p_i(l) = \frac{1}{t_{\max}} \sum_{t'=1}^{t_{\max}} \delta(x_i^{t'}, l)$
- Calculate the final segmentation from the winning label for each node: $x_i = \underset{l}{\operatorname{argmax}}\, p_i(l)$.
- Calculate the cut set C induced by x_i.
- Aggregate for each node i with edge $(i, j) \in C$, i.e. for all nodes i touching the segmentation border, the size of the subtrees h_i^t over all trials where the label x_i^t differed from the winning label x_i: $H_i = \sum_{t': x_i^t \neq x_i} h_i^{t'}$

Finally we propose another non-local alternative. Algorithm 3.3.1 takes advantage of an important property of the randomization of edge weights: the changing segmentation boundary (cut set $C(T^t)$) that results from each trial t of the stochastic watershed

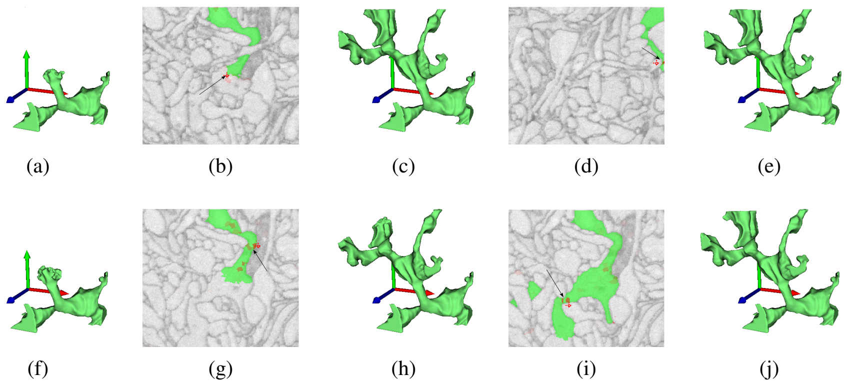


Figure 3: User guidance example for two of the proposed estimators. The top row displays the stochastic watershed using the stochastic segmentation instabiltiy estimator (Algorithm 3.3.1) , the bottom row the stochastic link instability estimator (Section 3.3.2). Displayed from left to right are the initial segmentation and two refinements based on seeding at the position of highest uncertainty (uncertainty is indicated by red color, the position of highest uncertainty by an arrow).

cut. This effect can be incorporated into an uncertainty estimator which attributes the magnitude of the aggregated segmentation boundary movement throughout the trials to individual edges.

The intuition behind the *stochastic segmentation instability* measure H_i is that very unstable segmentation boundaries indicate ambiguity in the data and need user verification. The definition of H_i (Algorithm 3.3.1) ensures that nodes receive high uncertainty when their label differs frequently from the winning label, or the affected subtrees are large. Nodes of highest criticality exhibit both problems.

4 Evaluation

4.1 Robot user

To evaluate the proposed uncertainty estimators objectively, we have designed an interactive segmentation robot [21]. The automaton tries to segment all neural processes in the data using two different seeding strategies. In the **ground truth strategy** the robot places two initial seeds, one inside the object of interest, one outside and then loops until convergence:

1. Calculate the set differences between ground truth and current segmentation.

2. Place a correcting single voxel seed in the center (maximum of the Euclidean distance transform) of the largest false positive or false negative region.

3. Re-run the segmentation algorithm with the new set of seeds.

Note that this segmentation robot requires knowledge of the complete three-dimensional ground truth for each iteration. This is clearly unrealistic, because if this knowledge were so readily available, then interactive segmentation would not be required in the first place.

The **uncertainty query strategy**, on the other hand, does not require full knowledge of the entire ground truth at each step. The robot begins by placing two initial seeds, one inside the object of interest, and one outside. It then loops until convergence:

1. Query the segmentation algorithm for the most uncertain region, using one of the confidence measures defined in Section 3.
2. Query the ground truth for the true label at the corresponding position.
3. Place a suitable seed at that position and re-run the segmentation algorithm.

4.2 Experiments

To evaluate the proposed uncertainty estimators for the seeded watershed cut we compare the estimators on a 3D segmentation problem from the neurosciences in a user guided segmentation setting. The nearly isotropic and densely annotated ground truth data is a subset of $400 \times 200 \times 200$ voxels from a 2000^3 volume of neural tissue acquired with a serial blockface electron microscopy (SBEM [22], Figure 1) and a $900 \times 450 \times 450$ densely annotated subset from a 2000^3 volume acquired with focused ion beam electron microscopy (FIBSEM [23], Figure 1). The reconstruction of the neural processes in this tissue is a segmentation problem that exhibits many properties that make it suitable for the seeded watershed cut [6].

We have tested all proposed uncertainty estimators with the *uncertainty query strategy* of the robot user against the *ground truth strategy*, which can be seen as an upper bound labeling strategy. During the segmentation process we recorded the resulting segmentation f-measure after each additional seed to compare the convergence rate of the robot for the different uncertainty estimators. Figure 4 shows the median across all neural processes in the respective ground truth. The parameters, namely the *bias* of the background seed (a background seed preference, [6]) and the amount of *perturbation* β in the case of the estimators based on the *stochastic watershed cut* were determined by a grid search over a training set consisting of 10% of the neural processes. For simplicity, the edge weights for the trials t were sampled as $w_{ij}^t \sim \mathrm{Unif}(w_{ij}, (1.0+\beta) * w_{ij})$. These edge weight distributions are simplistic, and even better results may be obtained when using more appropriate distributions. Figure 5 b displays the averaged standard deviation for $p_i(l)$ over 100 runs of the stochastic watershed cut with different trial counts t_{max}. While the average standard deviation does not converge for the trial counts considered here, Figure 5 a indicates, that the number of randomly sampled graphs has virtually no effect on the predictive quality of the two proposed stochastic uncertainty estimators, already a trial size of $t_{\text{max}} = 5$ can be used for successful user guidance. Thus our estimators incur only a small additional computational overhead compared to the standard watershed cut, which can be calculated in $\mathcal{O}(n\alpha(n))$ [24], i.e. quasi linear time.

5 Conclusion

We have presented and evaluated several novel uncertainty estimators for the seeded watershed cut. The proposed estimators are based on a minimum perturbation principle and stochastic edge weights, respectively. The proposed estimators were evaluated on a 3D biological neuroimaging application example that can profit from good uncertainty estimates and which exhibits many properties that make the seeded watershed cut a suitable algorithm. We showed that the proposed non-local uncertainty estimators yield a tremendous improvement in the number of user interactions compared to a simple local margin based approach which fails to query for more informative labels after the first few iterations. The proposed non-local estimators yield segmentation improvements that come close to an error correction strategy that relies on complete knowledge of the ground truth while incurring only an insignificant overhead compared to a standard watershed cut. Code and a GUI are made available at http://www.ilastik.org/carving-user-guidance.

6 Acknowledgements

FAH would like to gratefully dedicate this work to Prof. W. F. van Gunsteren on the occasion of his 65th birthday.

7 Appendix

7.1 General

Our uncertainty measures estimate by how much the segmentation would change if the edges in the spanning tree were replaced, and how likely these replacements are. MST edge exchange has been investigated in [25], and we repeat a useful lemma and definition from there:

Definition 7.1 (e, f exchange) *[25] Let T be a spanning tree of graph G. A e, f-exchange is a pair of edges, e, f where $e \in T, f \notin T$, and $T \setminus e \cup f$ is a spanning tree. The weight of the exchange e, f is $w(f) - w(e)$. The weight of tree $T \setminus e \cup f$ is the weight of tree T plus the weight of the exchange e, f.*

Lemma 7.2 *[25] A spanning tree T has minimum weight if and only if no e, f-exchange has negative weight.*

7.2 Link instability

Since we are only interested in changes of the MST that cause changes in the induced segmentation, we analyze the sufficient and neccessary conditions that yield a different segmentation:

Lemma 7.3 *An e, f-exchange resulting in a spanning tree $T' = T \setminus e \cup f$ induces a watershed segmentation different from T if and only if $f \in C(T)$.*

Proof *$\rightarrow$: Let e, f be an exchange with edge $e = (i, j) \in T$ and edge $f = (k, l) \in C$. Then, either node k or node l change their segmentation: before the exchange we obtain from the definition of the cut set C and $f \in C$: $x_k \neq x_l$, while after the exchange $f \in T$ and thus $x_k = x_l$.*

$\leftarrow$: Let e, f be an exchange with edge $e = (i, j) \in T$ and edge $f = (k, l)$. Assume $f \notin C$. First consider f to connect two nodes in the same subtree T^{-l} of e, i.e. $x_i = x_j = x_k = x_l = l$. Thus an e, f exchange will not change the segmentation of any node. Now consider f to connect two nodes in a different subtree $T^{-l'} \neq T^{-l}$ then e, i.e. $x_i = x_j = l$ and $x_k = x_l = l'$, then the e, f exchange will not produce a valid spanning tree: subtree $T^{-l'}$ contains a cycle, and subtree T^l is partitioned into two components.

Relying on the notion of an e, f-exchange (Definition 7.1) we now proof the correctness of the edge *link instability* Algorithm:

Lemma 7.4 *Algorithm 3.2.1 counts how often an edge in the minimum spanning tree is exchange partner in negative e, f exchanges resulting from all minimal single edge weight perturbations (Definition 3.4) that induce a different seeded watershed cut segmentation (Lemma 7.3).*

Proof *Item 1: When considering all segmentation changing perturbations involving a single edge, it suffices to consider the e, f-exchanges where $e \in T$ and $f \in C(T)$. This follows from Lemma 7.3.*

Item 2: When considering the minimal perturbations that move edge $f \in C(T)$ into the minimum spanning tree via an e, f exchange, it suffices to consider the edges on the path from node i or node j to the root node v_0, with edge $f = (i, j)$: if the edge e were not on a path from node i or node j to the root node v_0, with edge $f = (i, j)$, exchanging e with f would lead to a cycle. The fact that e has to be the edge of maximum weight on either path follows from Definition 3.4.

From Item 1 and Item 2 follows that the algorithm is correct.

7.3 Stochastic watershed cut

While the computational complexity of the stochastic watershed cut is fully treated in the supplementary material we provide some introductory remarks. To calculate $p_i(l)$, i.e. the probability of node v_i being assigned label $l \in L$ it is necessary to calculate the probability of v_i being in a subtree of the MST that is a child of a seed node v_{-l} with label l. Since even the calculation of the expected length of a minimum spanning tree of a stochastic graph is known to be a $\#P$-hard problem [26] it seems plausible that the calculation of the probability of two nodes belonging to the same subtree is as hard.

A proof of the #P-hardness for the stochastic watershed cut problem can be found in the supplementary material, what follows is an outline of the construction and the reduction to the $l - m$ network reliability problem.

Definition 7.5 ($l - m$ network reliability problem) *The two terminal network reliability problem is defined on an undirected graph $G(E, V)$ with edge weights $w_{ij} \sim Bernoulli(p_{ij})$ where p_{ij} is the probability of the connection between i and j being active (whereas an inactive edge is equivalent to a non-existing edge). The two terminal network reliability problem is then to calculate the probability of the existence of a*

path between two nodes l and m.
We reduce the two terminal network reliability problem for G to the stochastic watershed cut on a new graph $G'(V', E')$ that contains all edges and vertices of the original Graph G: We introduce a new node z' which is connected with weight $w'_{z'i'} = \alpha$ to all nodes $i' \in V'$ whith $\alpha \gg 1$. The edge weights $w'_{z'i'}$ connecting the newly introduced vertex z' to all nodes are set to α with probability 1 and ensure that the graph is completely connected and a MST can be constructed. The stochastic watershed cut edge weight distribution of the original edges w_{ij} is modeled as $w'_{i'j'} \sim Bernoulli(1 - p_{ij}) * \beta + 1, i \neq z'$ with $\beta \gg \alpha$.

Lemma 7.6 *The probability of the node m' being a child of z' in the $MST(G')$ is exactly the probability of a connection between m and l in the original graph G. Thus the two terminal network reliability problem can be reduced to a stochastic watershed cut.*

Lemma 7.7 *The stochastic watershed cut with arbitrary edge weight distributions PD_{ij} can be reduced to a series of two terminal network reliability problems.*

From Lemma 8.2 and Lemma 9.5 we conclude that the stochastic watershed cut is at least as hard as the two terminal network reliability problem which is known to be a $\#P$-hard problem [27].

Part I
Supplementary Material

In the supplementary material the proofs for Lemma 7.6 and Lemma 7.7 from the Appendix of the paper are given. We refer to the original Lemma 7.6 as Lemma 8.2 and Lemma 7.7 is Lemma 9.5 respectively.

8 Reducing the l, m network reliability problem to a stochastic mst problem

First we give a definition of the $l - m$ network reliability problem [27].

Definition 8.1 (l-m Network reliability problem) *The two terminal network reliability problem is defined on an undirected graph $G(E, V)$ with edge weights $w_{ij} \sim Bernoulli(p_{ij})$ where p_{ij} is the probability of the connection between i and j being active (whereas an inactive edge is equivalent to a non-existing edge). The two terminal network reliability problem is then to calculate the probability of the existence of a path between two nodes l and m.*

Intuition We reduce the $l-m$ network reliability problem to a stochastic minimum spanning tree calculation by constructing a new graph G' based on G. We add a new root vertex 0 to G' and introduce two label vertices -1 and -2. Graph G' is constructed in such a way, that when a connection between l and m exists in G node m is a child of vertex -1 in a MST of G' and a child of vertex -2 in a MST of G' if no connection exists in the original graph G.

The new graph $G'(V', E')$ contains all edges and vertices of the original Graph G. In addition a root vertex 0 is introduced which is connected with zero edge weight $w_{-1\;0} = 0, w_{-2\;0} = 0$ to the new label vertices -1 and -2. Node -1 is connected to node l' with zero edge weight $w_{-1\;-2} = 0$. Thes zero edge weights ensure that the corresponding edges are included in any MST of G'.

In addition the newly introduced vertex -2 is connected with weight $w'_{-2\;i'} = \alpha,\ \alpha > 1$ to all nodes $i' \in V'$ that are also present in G.

The edge weight distributions of the original edges $w_{i\;j}$ of G are modeled as

$$w'_{i'\;j'} \sim Bernoulli(1 - p_{ij}) * \beta + 1, \beta > \alpha$$

Thus a random trial in G that removes an edge (i, j) corresponds to an edge with weight $w'_{i'j'} = \beta + 1$ in G'. A random trial which leaves edge (i, j) intact in G induces an edge with weight $w'_{i'j'} = 1$ in G'.

Lemma 8.2 *The probability of the node m' being a child of node -1 in the $MST(G')$ is exactly the probability of a connection between m and l in the original graph G. Thus the two terminal network reliability problem can be reduced to a stochastic watershed cut.*

Proof *Consider a random draw of the edge weights. First we consider the case that the realized graph induced by the trial leaves l and m* connected *in G. It is easy to see that in this case m' must be a child of node -1 in the MST of the corresponding realization of G', since any spanning tree in which m' is a child of vertex -2 must include an edge $w'_{-2\;i'} = \alpha > 1$, while the* connectedness *of l and m in the original graph G implies by construction that a path $P(l', m')$ in G' exists with edge weights $w'_{i'\;j'} = 1, (i', j') \in P(l', m')$. Thus any MST in G' with root node 0 will have node m' and l' in a subtree of node -1 (which is by construction connected with zero edge weight to l').*

Secondly we consider the case that the realized graph induced by the trial leaves l and m disconnected*. It is also easy to see that in this case m' must be a child of node -2 in any MST of the corresponding realization of G', since any spanning tree connecting m' to node -1 includes an edge $w'_{i'j'} = \beta > \alpha$ since the* disconnectedness *in G implies that any $P(l', m')$ in G' includes at least on edge of such weight (by construction of the edge weights in G' which assigns weight $w'_{i'j'} = \beta$ when the bernoulli trial in the original graph G removes edge (i', j')). Thus any minimum spanning tree connects node m' to node -2 since this incurs the cheaper cost of $w'_{-2\;m'} = \alpha < \beta$.*

We showed that any random trial that leaves l and m connected in G implies that m' is a child of node -1 in the $MST(G')$, while any random trial that disconnects l and m in G implies that m' is a child of -2 in the $MST(G')$.

Since the final probability is defined by the outcome of all possible trials and the outcomes are linked in the described way it has been shown that the $l - m$ terminal network reliability can be answered by calculating the probability for m' being a child of a newly introduced node -1 in a MST of a newly constructed Graph G'. This is a stochastic watershed cut problem.

9 Solving the stochastic mst problem with a series of l, m network reliability problems

Intuition We reduce the stochastic mst problem on a graph G with root node 0 and two seed type nodes -1 and -2 to a series of $l - m$ network reliability problems by calculating the probability for a path $P(-1, j)$ between seed type node -1 and node j with height $H(P(-1, j)) \leq h$. This can be done by solving a network connectivity problem on a new graph and considering the cumulative probability distributions $CDF_{ij}(h)$ instead of $PD_{ij}(h)$. We denote the event that a path from node -1 to node j with height equal or lower then h exists by event

$$A_{-1\ j, \leq h}$$

The probability that a path $H(P(-1, j)) \leq h$ exists while at the same time no path of such weight exists from seed node -2 is given by the probability for both events happening:

$$p(A_{-1\ j, \leq h} \wedge A_{-2\ j, >h})$$

By integrating over h the overall probability for the existance a lower path from node -1 to node j then from node -2 is obtained (this event is denoted as $B_{-1\ j<-2\ j}$):

$$p(B_{-1\ j<-2\ j}) = \int_{0}^{\infty} p(A_{-1\ j, \leq h} \wedge A_{-2\ j, >h}) dh$$

The existance of a lower path from -1 to j guarantess that node j is a child of node -1 in a MST of G' with root node 0 (Lemma 9.3). Thus by calculating $p(B_{-1\ j<-2\ j})$, i.e. the probability for j being a child of -1, the solution to a stochastic watershed cut problem is obtained.

Definition 9.1 (Height of a path $H(P(i, j))$) *The height of a path $H(P(i, j))$ from i to j in G is defined as the maximum edge weight on this path:*

$$H(P(i, j)) = \max_{w_{kl} \in P(i,j)} w_{kl}$$

Definition 9.2 (Lowest path $P'(i, j)$) *The Lowest possible path $P'(i, j)$ from i to j in G is defined as the path with the lowest maximum edge weight amongs all possible paths from i to j $\mathcal{P}(i, j)$:*

$$P'(i, j) = \underset{P(i,j) \in \mathcal{P}(i,j)}{\operatorname{argmin}} H(P(i, j))$$

Lemma 9.3 *Let i be a node in G and $P'(-1, i)$ be a lower path then $P'(-2, i)$, i.e. $H(P'(-1, i)) < H(P'(-2, i))$, then node i is a child of seed node -1 in a minimum spanning tree.*

Proof *Assuming the contrary violates the minimum spanning tree property of no negative edge exchanges.*

Lemma 9.4 *Let G' be a graph constructed from G by setting $w'_{i'j'} \sim Bernoulli(CDF_{ij}(h))$ where CDF_{ij} is the cumulative probability density function of the edge weight distribution PD_{ij}. The probability of a path $P(-1,i)$ with maximum height $H(P(-1,i)) \leq h$ is equal to the probability of a connection between node -1 and node i in G'.*

Proof *This follows from the definition of the edge weights $w'_{i'j'} \sim Bernoulli(CDF_{ij}(h))$ and the definition of the cumulative probability density function. I.e. $Bernoulli(CDF_{ij}(h))$ expresses the probability that edge weight $w_{ij} \leq h$. The probability for a path $P_{G'}(-1,i)$ in G' is then the probability of a $P_G(-1,i)$ in G such that all $w_{nm} \leq h, w_{nm} \in P_G(-1,i)$.*

Lemma 9.5 *Let G be a graph with edge weight $w_{ij} \sim PD_{ij}$, root vertex v_0 and two label type vertices v_{-1} and v_{-2}. The stochastic watershed cut problem, i.e. the calculation of the probability of node i being a child of node -1 in a MST with root node 0 can be reduced to a series of $l-m$ network reliability problems.*

Proof *Let G' be a graph constructed from G by setting $w'_{i'j'} \sim Bernoulli(CDF_{ij}(h))$ where CDF_{ij} is the cumulative probability density function of the edge weight distribution PD_{ij}. From Lemma 9.4 we have that by setting $w'_{-2\ j} = 0$ and $w'_{-1i} = w_{-1i}$ and solving the two terminal network reliability problem between node -1 and node j one obtains the probability for a path from -1 to j with maximum height $\leq h$*

$$p(A_{-1\ j,\leq h})$$

the existence of such a path is denoted as event $A_{-1\ j,<h}$. The probability for the events $p(A_{-2\ j,\leq h})$ is calculated accordingly. We note that $p(A_{-1\ j,>h}) = 1 - p(A_{-1\ j,\leq h})$.

Let G'' be a graph constructed from G by setting $w''_{ij} \sim Bernoulli(CDF_{ij}(h))$ where CDF_{ij} is the cumulative probability density function of the edge weight distribution PD_{ij}. By setting $w''_{-2\ i} = w_{-2\ i}$ and $w''_{-1\ i} = w_{-1\ i}$ and connecting node -1 in G'' to the other seed node -2, i.e. $w''_{-1\ -2} = 1$ one can calculate the probability of a path from -1 or -2 to j with maximum height $\leq h$

$$p(A_{-1\ j,\leq h} \vee A_{-2\ j,\leq h}) =$$

$$p(A_{-1\ j,\leq h}) + p(A_{-2\ j,\leq h}) - p(A_{-1\ j,\leq h} \wedge A_{-2\ j,\leq h})$$

Thus, $p(A_{-1\ j,\leq h} \wedge A_{-2\ j,\leq h})$ can be calculated by solving three network reliability problems.

From

$$p(A_{-1\ j,\leq h} \wedge A_{-2\ j,\leq h}) = p(A_{-1\ j,\leq h}) p(A_{-2\ j,\leq h} | A_{-1\ j,\leq h})$$

and by using

$$p(A_{-2\ j,>h} | A_{-1\ j,\leq h}) = 1 - p(A_{-2\ j,\leq h} | A_{-1\ j,\leq h})$$

one obtains

$$p(B_{-1\ j<-2\ j}(h)) := p(A_{-1\ j,\leq h} \wedge A_{-2\ j,>h}) =$$

$$p(A_{-1\ j,\leq h}) p(A_{-2\ j,>h} | A_{-1\ j,\leq h})$$

By integrating over h the probability for the existance of a lower path from -1 to j then from -2 to j is calculated as:

$$p(B_{-1\ j<-2\ j}) = \int_{0}^{\infty} p(A_{-1\ j,\leq h} \wedge A_{-2\ j,>h})dh$$

which in addition with Lemma 9.3 gives the probability of node j being a child of node -1 in a MST with root node 0.

We conclud that a stochastic watershed cut problem can be reduced to a series of network reliability problems.

In Lemma 8.2 it was shown that any l-m network reliability problem can be expressed as a stochastic watershed cut problem. In Lemma 9.5 we have shown how to decompose a stochastic watershed cut problem with arbitrary edge weight distributions PD_{ij} into a series of l-m network reliability problems. We conclude that the stochastic watershed cut is at least as hard as the two terminal network reliability problem which is known to be a $\#P$-hard problem [27].

References

[1] A. Fathi, M. Balcan, X. Ren, and J. Rehg, "Combining self training and active learning for video segmentation," BMVC, 2011.

[2] B. Settles, "Active learning literature survey," tech. rep., University of Wisconsin-Madison, 2009.

[3] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen, "icoseg: Interactive co-segmentation with intelligent scribble guidance," in *CVPR*, 2010.

[4] A. Top, G. Hamarneh, and R. Abugharbieh, "Active learning for interactive 3d image segmentation," *MICCAI*, 2011.

[5] A. Lucchi, K. Smith, R. Achanta, G. Knott, and P. Fua, "Supervoxel-based segmentation of em image stacks with learned shape features," 2010.

[6] C. N. Straehle, U. Köthe, G. Knott, and F. A. Hamprecht, "Carving: Scalable interactive segmentation of neural volume electron microscopy images," in *MICCAI*, 2011.

[7] J. Cousty, G. Bertrand, L. Najman, and M. Couprie, "Watershed cuts: minimum spanning forests and the drop of water principle," *IEEE PAMI*, pp. 1362–1374, 2009.

[8] C. Couprie, L. Grady, L. Najman, and H. Talbot, "Power watershed: A unifying graph-based optimization framework," *IEEE PAMI*, 2010.

[9] R. Lotufo and W. Silva, "Minimal set of markers for the watershed transform," in *Proceedings of International Symposium on Mathematical Morphology*, 2002.

[10] L. Grady, "Random walks for image segmentation," *IEEE PAMI*, 2006.

[11] Y. Boykov and M. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in nd images," in *ICCV*, 2001.

[12] C. Chefd'hotel and A. Sebbane, "Random walk and front propagation on watershed adjacency graphs for multilabel image segmentation," 2007.

[13] L. Grady and A. Sinop, "Fast approximate random walker segmentation using eigenvector precomputation," in *CVPR*, 2008.

[14] O. Juan and Y. Boykov, "Active graph cuts," in *CVPR*, 2006.

[15] G. Papandreou and A. Yuille, "Perturb-and-map random fields: Using discrete optimization to learn and sample from energy models," in *ICCV*, 2011.

[16] P. Kohli and P. Torr, "Measuring uncertainty in graph cut solutions–efficiently computing min-marginal energies using dynamic graph cuts," *ECCV*, 2006.

[17] J. Angulo and D. Jeulin, "Stochastic watershed segmentation," in *Intern. Symp. on Mathematical Morphology*, 2007.

[18] F. Meyer and J. Stawiaski, "A stochastic evaluation of the contour strength," in *DAGM*, 2010.

[19] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE PAMI*, 1991.

[20] C. Chen, D. Freedman, and C. Lampert, "Enforcing topological constraints in random field image segmentation," in *CVPR*, 2011.

[21] H. Nickisch, C. Rother, P. Kohli, and C. Rhemann, "Learning an interactive segmentation system," in *Indian Conference on Computer Vision, Graphics and Image Processing*, 2010.

[22] W. Denk and H. Horstmann, "Serial Block-Face scanning electron microscopy to reconstruct Three-Dimensional tissue nanostructure," *PLoS Biology*, 2004.

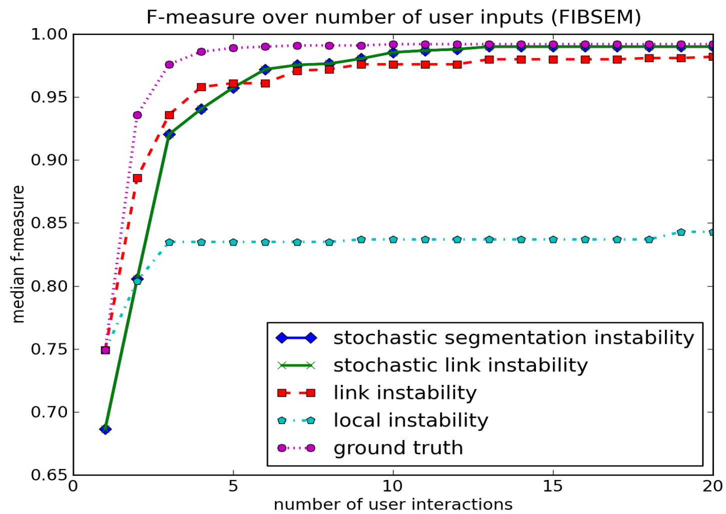
[23] G. Knott, H. Marchman, D. Wall, and B. Lich, "Serial section scanning electron microscopy of adult brain tissue using focused ion beam milling," *The Journal of Neuroscience*, 2008.

[24] B. Chazelle, "A minimum spanning tree algorithm with inverse-ackermann type complexity," *Journal of the ACM*, 2000.

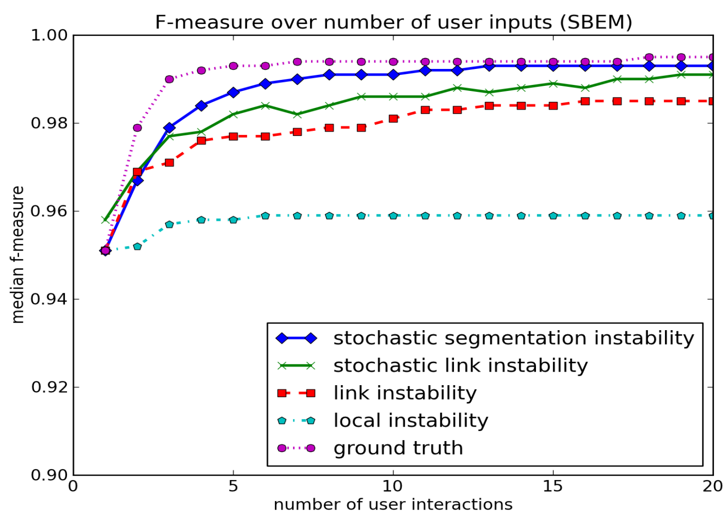
[25] H. Gabow, "Two algorithms for generating weighted spanning trees in order," *SIAM Journal on Computing*, 1977.

[26] P. Kamousi and S. Suri, "Stochastic minimum spanning trees and related problems," 2011.

[27] H. Bodlaender and T. Wolle, "A note on the complexity of network reliability problems," *IEEE Trans. Inf. Theory*, 2004.

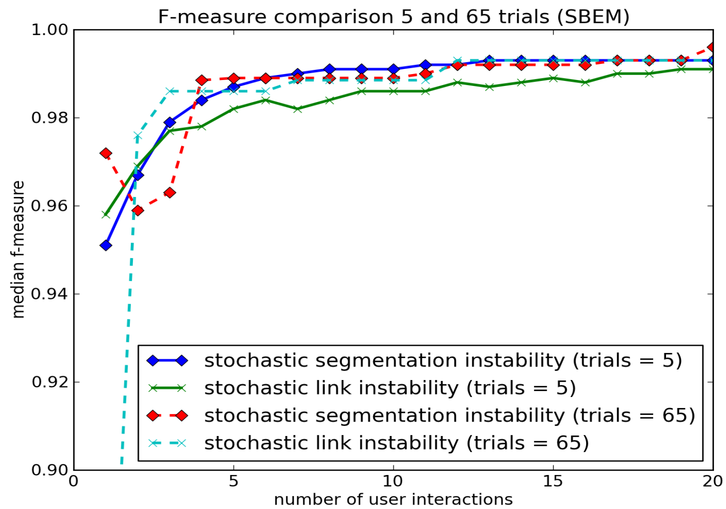


(a)

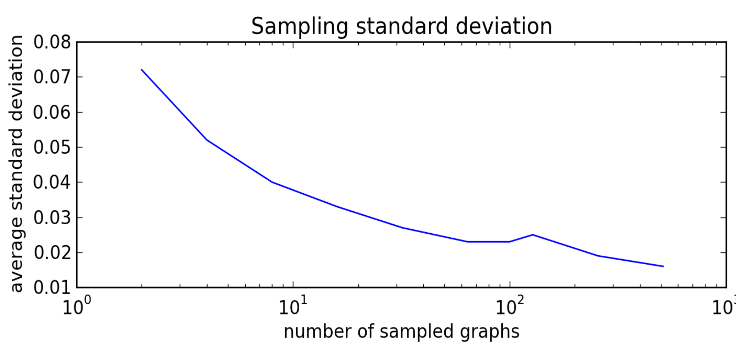


(b)

Figure 4: Median F-measure over number of user interactions using the different uncertainty estimators as the query strategy for the segmentation robot for the two different datasets. (a) FIBSEM. (b) SBEM.



(a)



(b)

Figure 5: (a) Median F-measure over number of user interactions using 5 and 65 randomly sampled graphs. (b) Averaged standard deviation for $p_i(l)$ using 100 runs of the stochastic watershed over the number of randomly sampled Graphs along the x-axis.