

The Semantic Mutex Watershed for Efficient Bottom-Up Semantic Instance Segmentation

Steffen Wolf^{1*}
 Alberto Bailoni¹

Yuyan Li^{1*}
 Anna Kreshuk²

Constantin Pape²
 Fred A. Hamprecht¹

¹HCI/IWR, Heidelberg University, Germany

²EMBL Heidelberg, Germany

Abstract

Semantic instance segmentation is the task of simultaneously partitioning an image into distinct segments while associating each pixel with a class label. In commonly used pipelines, segmentation and label assignment are solved separately since joint optimization is computationally expensive. We propose a greedy algorithm for joint graph partitioning and labeling derived from the efficient Mutex Watershed partitioning algorithm [42]. It optimizes an objective function closely related to the Symmetric Multiway Cut objective and empirically shows efficient scaling behavior. Due to the algorithm's efficiency it can operate directly on pixels without prior over-segmentation of the image into superpixels. We evaluate the performance on the Cityscapes dataset (2D urban scenes) and on a 3D microscopy volume. In urban scenes, the proposed algorithm combined with current deep neural networks outperforms the strong baseline of 'Panoptic Feature Pyramid Networks' by Kirillov et al. (2019). In the 3D electron microscopy images, we show explicitly that our joint formulation outperforms a separate optimization of the partitioning and labeling problems.

1. Introduction

Image segmentation literature distinguishes *semantic segmentation* - associating each pixel with a class label - and *instance segmentation*, i.e. detecting and segmenting individual objects while ignoring the background. The joint task of simultaneously assigning a class label to each pixel and grouping pixels to instances has been addressed under different names, including semantic instance segmentation, scene parsing [39], image parsing [40], holistic scene understanding [43] or instance-separating semantic segmentation [28]. Recently, a new metric and evaluation approach to such problems has been introduced under the name of

panoptic segmentation [19].

From a graph theory perspective, semantic instance segmentation corresponds to the simultaneous partitioning and labeling of a graph. Most greedy graph partitioning algorithms are defined on graphs encoding attractive interactions only. Clusters are then formed through agglomeration or division until a user-defined termination criterion is met (often a threshold or a desired number of clusters). These algorithms perform pure instance segmentation. The semantic labels for the segmented instances need to be generated independently.

If repulsive - as well as attractive - forces are defined between the nodes of the graph, partitioning can be formulated as a Multicut problem [2]. In this formulation clusters emerge naturally without the need for a termination criterion. Furthermore, the Multicut problem can be extended to include the labeling of the graph, delivering a semantic instance segmentation from a joint optimization of partitioning and labeling [24]. The main drawback of this formulation is that the Multicut problem is NP-hard.

We propose to solve the joint partitioning and labeling problem by an efficient algorithm which we term Semantic Mutex Watershed (SMW), inspired by the Mutex Watershed [41]. In more detail, in this contribution we:

- propose a fast algorithm for joint graph partitioning and labeling
- prove that the algorithm minimizes (exactly) an objective function closely related to the Symmetric Multiway Cut objective
- demonstrate competitive performance on natural and biological images.

2. Related Work

Semantic segmentation. State-of-the-art semantic segmentation algorithms are based on convolutional neural networks (CNNs) which are trained end-to-end. The networks commonly follow the design principles of image classification networks (*e.g.* [16, 38, 23]), replacing the fully con-

*Equal contribution.

nected layers at the end with convolutional layers to form a fully convolutional network [30]. This architecture can be further extended to include encoder-decoder paths [37], dilated or atrous convolutions [45, 6] and pyramid pooling modules [7, 46].

Instance segmentation. Many instance segmentation methods use a detection or a region proposal framework as their basis; object segmentation masks are then predicted inside region proposals. A cascade of multiple networks is employed by [12], each solving a specific subtask to find the instance labeling. Mask-RCNN [15] builds on the bounding box prediction capabilities of Faster-RCNN [36] to simultaneously produce masks and class predictions. An extension of this method with an additional semantic segmentation branch has been proposed in [18] as a single network for semantic instance segmentation.

In contrast to the region-based methods, proposal-free algorithms often start with a pixel-wise representation which is then clustered into instances [44, 21, 13]. Alternatively, the distance transform of instance masks can be predicted and clustered by thresholding [4].

Graph-based segmentation. Graph-based methods, used independently or in combination with machine learning on pixels, form another popular basis for image segmentation algorithms [14]. In this case, the graph is built from pixels or superpixels of the image and the instance segmentation problem is formulated as graph partitioning. When the number of instances is not known in advance and repulsive interactions are present between the graph nodes, graph partitioning can in turn be formulated as a Multicut or correlation clustering problem [2]. This NP-hard problem can be solved reasonably fast for small problem sizes with integer linear programming solvers [1] or approximate algorithms [34, 5]. A modified Multicut objective is introduced by [41] together with the Mutex Watershed - an efficient clustering algorithm for its optimization.

The Multicut objective can be extended to solve a joint graph partitioning and labeling problem [17, 24] for simultaneous instance and semantic segmentation. In practice, the computational complexity of the joint problem only allows for approximate solutions [28], possibly combined with reducing the problem size by over-segmentation into superpixels. This formulation has been applied to natural images by [20] and to biological images by [22].

Similar to the semantic segmentation use case, CNNs can be used to predict pixel and superpixel affinities which serve as edge weights in the graph partitioning problem [27, 31, 29].

3. The Semantic Mutex Watershed

The centerpiece of this paper, the Semantic Mutex Watershed algorithm, solves the semantic instance segmentation problem by jointly finding a graph partitioning and labeling. In this section we present the graph-based formulation of the semantic instance segmentation problem and define an objective function related to the Symmetric Multiway Cut problem [24]. Then we introduce the Semantic Mutex Watershed algorithm and prove that it can optimize this objective efficiently. Finally we show that the proposed objective constitutes a generalization of the Mutex Watershed Objective introduced in [41].

3.1. Joint Partitioning and Labeling of Graphs

Similar to instance segmentation algorithms, we build a graph of image pixels (voxels) or superpixels and formulate the semantic instance segmentation problem as joint partitioning and labeling of the graph.

Weighted graph with terminal nodes. For an undirected weighted graph $G = G(V, E, W)$ we refer to the nodes V as *internal nodes* and the edges E as *internal edges*. We differentiate between *attractive edges* E^+ and *repulsive edges* E^- that make up the internal edges $E = E^+ \cup E^-$. Each edge $e \in E$ is associated with a real-valued positive weight $w_e \in W = W^+ \cup W^-$. The weights encode the attraction and repulsion between the incident nodes of each edge. A large *attractive weight* $w_{uv} \in W^+$ encodes a high tendency for the nodes u and v to belong to the same partition element. Equivalently, a large *repulsive weight* $w_{uv} \in W^-$ indicates a strong inclination of u and v to belong to separate clusters.

Semantic instance segmentation can be achieved by clustering the internal nodes and assigning a semantic label $l \in \{l_0, \dots, l_k\}$ to each cluster. We extend G by k *terminal nodes* $\{t_0, \dots, t_k\} \in T$ where each t_i is associated with a label l_i . Every internal node $v \in V$ is connected to every t by a weighted *semantic edge* $e \in E^S$. Here, a large *semantic weight* $w_{ut} \in W^S \subseteq \mathbb{R}^+$ implies a strong association of internal node u with the label of the terminal node t . The extended graph thus becomes $G'(V', E', W')$ with $V' = V \cup T$, $E' = E \cup E^S$ and $W' = W \cup W^S$. Figure 1(a) shows a simple example of such an extended graph.

Algorithm 1: The Semantic Mutex Watershed algorithm. The differences to the Mutex Watershed are marked in blue.

```

Input: weighted graph  $G'(V \cup T, E', W')$ 
Output: clusters and labeling defined by  $A$ 
Initialization :  $A = \emptyset$ 
for  $(i, j) = e \in E'$  in descending order of  $w_e$  do
    if  $e \in E^+$  then
        if not mutex  $(i, j)$ 
        and not class  $(i) \neq \text{class}(j)$  then
            merge  $(i, j) : A \leftarrow A \cup e$ 
    else if  $e \in E^-$  then
        if not connected  $(i, j)$  then
            addMutex  $(i, j) : A \leftarrow A \cup e$ 
    else if  $e \in E^S$  then
        if class  $(i) = \emptyset$  or class  $(i) = l_j$  then
            assignLabel  $(i, j) : A \leftarrow A \cup e$ 
    return  $A$ 

```

Symmetric Multiway Cut. In the Symmetric Multiway Cut an optimal semantic instance segmentation of such a graph is formulated as a constrained energy minimization/integer linear program (ILP) [24]:

$$\max_{a \in \{0,1\}^{|E'|}} \sum_{e \in E'} w_e^p \mathbf{a}_e \quad (1)$$

$$\sum_{(i,j) \in P \cap E^-} \mathbf{a}_{ij} + \sum_{(i,j) \in P \cap E^+} (1 - \mathbf{a}_{ij}) \geq (1 - \mathbf{a}_{uv}) \quad (2)$$

$$\forall (u, v) \in E^+; \forall P \in \text{Path}(u, v) \subseteq E$$

$$\sum_{(i,j) \in P \cap E^-} \mathbf{a}_{ij} + \sum_{(i,j) \in P \cap E^+} (1 - \mathbf{a}_{ij}) \geq \mathbf{a}_{uv} \quad (3)$$

$$\forall (u, v) \in E^-; \forall P \in \text{Path}(u, v) \subseteq E$$

$$\sum_{t \in T} \mathbf{a}_{it} = 1 \quad \forall i \in V \quad (4)$$

$$(1 - \mathbf{a}_{uv}) \geq \mathbf{a}_{vt} - \mathbf{a}_{ut} \quad \forall (u, v) \in E^+; t \in T \quad (5)$$

$$(1 - \mathbf{a}_{uv}) \geq \mathbf{a}_{ut} - \mathbf{a}_{vt} \quad \forall (u, v) \in E^+; t \in T \quad (6)$$

where $p = 1$. The segmentation consistency is ensured by the cycle inequalities (2) and (3). Equation (4) enforces that every node is uniquely assigned to a terminal node. Equations (5) and (6) ensure the consistency between partition labeling and semantic labeling. A detailed discussion of the objective's properties will follow in section 3.3 and the relation to the objective in [24] is derived in appendix A. Although this is *in general* a hard optimization problem, we will show that for sufficiently large p this objective can be solved exactly and efficiently by the algorithm introduced in the next section.

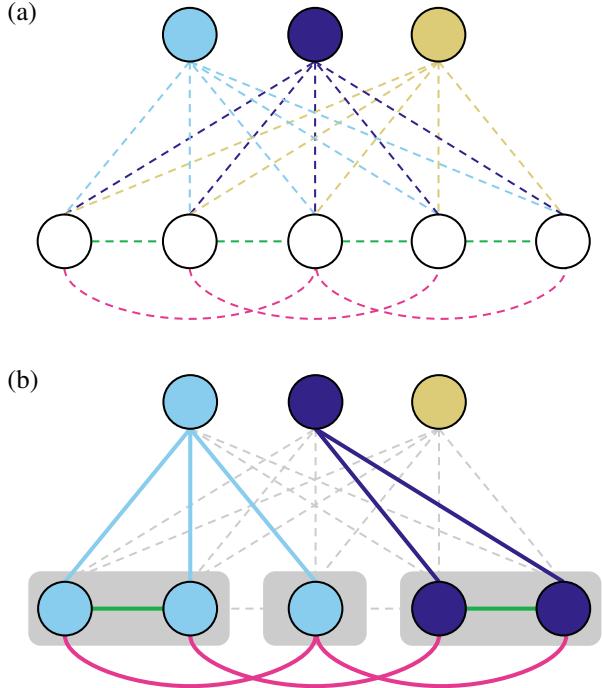


Figure 1: (a) Example of an extended graph. Nodes on top are terminal nodes with each color representing a label class. The associated semantic edges are colored correspondingly. The internal nodes are on the bottom with attractive (green) and repulsive (red) edges between them. (b) Semantic instance segmentation. Edges that are part of the active set are shown in bold. Note that two adjacent nodes with the same label are not necessarily clustered together.

3.2. The Semantic Mutex Watershed Algorithm.

We will now introduce a simple algorithm that greedily constructs a solution to eq. (1). Although this will most likely not be an optimal solution to the NP-hard Symmetric Multiway Cut in general, we will show in section 3.3 that it becomes optimal when p is large.

The clustering and label assignment of G is described by a set of *active* edges, which are chosen by the algorithm: $A \subseteq E'$ where $A \cap E^+$, $A \cap E^-$ and $A \cap E^S$ encode merges, mutual exclusions and label assignments, respectively. In order to restrict A to a consistent partitioning and labeling we will make the following definitions:

We define two internal nodes $i, j \in V$ as connected if they are connected by active attractive edges, i.e.

$$\text{connected } (i, j) \Leftrightarrow \exists \pi_{i \leadsto j} \subseteq A \cap E^+ \quad (7)$$

Here $\pi_{i \leadsto j}$ denotes a path from node i to node j . We also define the mutual exclusion between two nodes as

$$\text{mutex } (i, j) \Leftrightarrow \exists \pi_{i \leadsto j} \subseteq A \text{ with } |\pi \cap E^-| = 1. \quad (8)$$

Two nodes are thus mutual exclusive if they are connected by a path from i to j with exactly one repulsive edge. Furthermore, a label l_j is assigned to a node i if this node is connected to the corresponding terminal node t_j by attractive and semantic edges:

$$\text{class}(i) = l_j \Leftrightarrow \exists \pi_{i \leadsto t_j} \subseteq A \cap (E^+ \cup E^S). \quad (9)$$

For unlabeled nodes we use the notation $\text{class}(i) = \emptyset$.

Algorithm. The Semantic Mutex Watershed algorithm is an extension of the Mutex Watershed algorithm introduced by [42]. It augments the partitioning of the latter with a consistent labeling. The algorithm is shown in [algorithm 1](#) with the additions to [42] highlighted. In the following we explain the syntax and procedure of the shown pseudocode.

All edges E' are sorted in descending order of their weight and put in a priority queue. While traversing the queue, the decision to add an edge to the set A is made depending on the type of edge:

Attractive edges: The edge is added if the incident nodes are not mutual exclusive and not labeled differently.

Repulsive edges: The edge is added if the incident nodes are not connected.

Semantic edges: The edge is added if the node is either unlabeled or already has the same label as the edge's terminal node.

Following these rules, the set of attractive edges in the final set $A \cap E^+$ form clusters in the graph, which are each connected to a single terminal node indicating the labeling. [Figure 1\(b\)](#) shows a simple example of such an active set.

Efficient Implementation with Maximum-Spanning-Trees. The SMW is similar to the efficient Kruskal's maximum spanning tree algorithm [25] and can feasibly be applied to pixel-graphs of large images and even image volumes. Our implementation utilizes an efficient union-find data structure, mutex relations are realized/searched through a hash table.

Mutex Watershed as Special Case. The Mutex Watershed algorithm is embedded in the Semantic Mutex Watershed as the special case when there are zero or one label ($|T| \in \{0, 1\}$).

3.3. The Semantic Mutex Watershed Objective

In this section we prove that the Semantic Mutex Watershed Algorithm solves the ILP objective in [eq. \(1\)](#) for sufficiently large (*dominant*) powers p . To this end, we will extend the proof of [41] by semantic edges. First, we will review the definitions of *dominant powers* and *mutex constraints*. Second, we introduce an additional set of constraints acting on semantic edges and use it to define the Semantic Mutex Watershed Objective as a relaxed version of the Symmetric Multiway Cut. Finally, we prove that the solution found by the SMW is indeed optimal.

Dominant Power. Let $\mathcal{G} = (V, E, W)$ be an edge-weighted graph, with unique weights $w_e \in \mathbb{R}_0^+$, $\forall e \in E$. We call $p \in \mathbb{R}^+$ a dominant power if:

$$w_e^p > \sum_{s \in E, w_s < w_e} w_s^p \quad \forall e \in E, \quad (10)$$

Note that there exists a dominant power for any finite set of edges, since for any $e \in E$ we can divide (10) by w_e^p and observe that the normalized weights w_s^p/w_e^p (and any finite sum of these weights) converges to 0 when p tends to infinity.

Semantic Mutex Watershed Constraints. To formalize the rules of the algorithm defined above, we first define special subsets of the active set A . First, the set of all cycles containing exactly one repulsive edge is defined as

$$\mathcal{C}_1(A) := \{c \in \text{cycles}(\mathcal{G}) \mid c \subseteq A \cap E \text{ and } |c \cap E^-| = 1\}. \quad (11)$$

We define the **mutex constraint** as requiring $\mathcal{C}_1(A) = \emptyset$, which is exactly the rule that two mutual exclusive nodes must not be connected.

Furthermore, we define the set $\mathcal{P}(A)$ of all paths π that connect two distinct terminal nodes through attractive and semantic edges:

$$\mathcal{P}(A) := \{\pi_{t \leadsto t'} \subseteq A \cap (E^+ \cup E^S) \mid t, t' \in T\} \quad (12)$$

The algorithm must never connect two terminal nodes through such a path, thus we define the **label constraint** $\mathcal{P}(A) = \emptyset$. This ensures the consistency between the partitioning and labeling. The mutex and label constraint are necessary but not sufficient to fulfill the linear constraints in [eqs. \(2\) to \(6\)](#) (the formal derivation can be found in [appendix B](#)).

Lemma 3.1 (Optimality of the Semantic Mutex Watershed). Let $G = G(V', E', W') = G(V \cup T, E \cup E^S, W \cup W^S)$ be an edge-weighted graph extended by terminal nodes T , with unique weights $w_e \in \mathbb{R}_0^+$ and $p \in \mathbb{R}^+$ a dominant power. Then the Semantic Mutex Watershed [Algorithm 1](#) finds the optimal solution to the integer linear program

$$\max_{a \in \{0, 1\}^{|E'|}} \sum_{e \in E'} w_e^p a_e \quad (13)$$

$$\text{s.t. } \mathcal{C}_1(A) = \emptyset, \quad (14)$$

$$\mathcal{P}(A) = \emptyset, \quad (15)$$

$$\text{with } A := \{e \in E \mid a_e = 1\}. \quad (16)$$

Proof. [41] show that for $T = \emptyset$ the SMW finds the optimal solution because it enjoys the properties *greedy choice* and *optimal substructure*. Their proof of optimal substructure does not rely on the specific constraints in the ILP. Thus it

can also be applied with the additional constraint in eq. (15), giving the ILP eqs. (13) to (16) optimal substructure.

In every iteration the SMW adds the feasible edge e with the largest weight to the active set. Due to the dominant power, its energy contribution is larger than for any combination of edges e' with $w'_e < w_e$. Thus, SMW has the greedy choice property [11]. It follows by induction that the SMW algorithm finds the globally optimal solution to the SMW objective. \square

We can now finally observe that the SMW algorithm always yields a consistent graph partitioning and labeling which fulfills the Symmetric Multiway Cut constraints. Thus, the Semantic Mutex Watershed algorithm returns an optimal solution of eqs. (1) to (6) if p is set to a dominant power. In particular, if $p = 1$ is dominant then the SMW solution is also an optimal solution to the Symmetric Multiway Cut.

4. Experiments

We will now demonstrate how to apply the SMW algorithm to semantic instance segmentation of 2D and 3D images. We start from showing how existing CNNs can be used as graph weight estimators and compare different sources of edge weights on the Cityscapes dataset. Additionally, we apply the SMW algorithm to a 3D electron microscopy volume and demonstrate its efficiency and scalability.

4.1. Affinity Generation with Neural Networks

The only input to the SMW are the graph weights; it does not require any hyperparameters such as thresholds. Consequently, its segmentation quality relies on good estimates of the graph weights $W' = W \cup W^S$. In this section we present how state-of-the-art CNNs can be used as sources for these weights.

Affinity Learning. Affinities are commonly used in instance segmentation; many modern algorithms train CNNs to directly predict pixel affinities. A universal approach is to employ a stencil pattern that describes for each pixel which neighbours to consider for the affinity computation. Regularly spaced, multi-scale stencil patterns are widely used for natural images [31, 29] and bio-medical data [42, 27].

The predicted affinities are usually in the interval $[0, 1]$ and can be interpreted as pseudo-probabilities. We use these affinities directly as weights for the attractive edges and invert them to get the repulsive edge weights.

Mask-RCNN produces overlapping masks that have to be resolved for a consistent panoptic segmentation. We achieve this with the SMW by deriving affinities from the foreground probabilities of each mask. A straightforward approach is to compute the (attractive) affinity $a(i, j)$ of two

pixels as their joint foreground probability, weighted by the classification score s : $a(i, j) = s p(i) p(j)$.

We find that sparse repulsive edges work well in practice, as they lead to faster inference and reduced over-segmentation on the instance boundaries. For this reason, we sample random points from all pairs of masks and add (repulsive) edges with weight proportional to a soft intersection over union of two masks m and n :

$$w_{nm} = 1 - \frac{\sum_{q \in V} p_m(q)p_n(q)}{\sum_{q \in V} \max(p_m(q), p_n(q))}. \quad (17)$$

Semantic Segmentation CNNs. State of the art CNNs [8, 46] achieve high quality results on semantic segmentation tasks. The output of the last softmax layer usually used in these networks can be interpreted as the normalized probability of each pixel belonging to each class. Thus, we can use these predictions directly as semantic weights W^S .

Additionally, we derive affinities from the stuff class probabilities; we treat each stuff class separately and again compute the affinity of two pixels as their joint probability of being in each stuff class c , i.e.: $a_c(i, j) = p_c(i)p_c(j)$. This cannot be done for thing classes since they can have multiple instances.

4.2. Panoptic Segmentation on Cityscapes

We apply the SMW on the challenging task of panoptic segmentation on the Cityscapes dataset [10]. We illustrate how the different sources of affinities can be used and combined and show their different strengths and weaknesses.

Dataset. The Cityscapes dataset consists of urban street scene images taken from a driver’s perspective. It has 5k densely annotated images separated into train (2975), val (500) and test (1525) set. Since there is no public evaluation server for panoptic segmentation on the test set, we report all results on the validation set. There are 19 classes with 11 stuff classes and 8 thing classes.

Implementation Details. We employ and combine multiple sources of graph weights to build the SMW graph. We train a Deeplab 3+ [8] network for semantic edge weight and affinities prediction following [29]. We employ the Mask-RCNN [15] implementation provided by [32] and train a model on Cityscapes following [15]’s training configuration. Further implementation details can be found in appendix C.1.

Study of Affinity Sources. We evaluate the semantic instance segmentation performance of the SMW in terms of the “panoptic” metric using different combinations of the graph weight sources discussed above. In table 1 we compare the PQ metric on the Cityscapes dataset.

The best performance can be achieved with a combination of Mask-RCNN affinities and Deeplab 3+ for semantic predictions outperforming the strong baseline of [18] listed



Figure 2: Semantic instance segmentation. *First three columns:* Results on Cityscapes using semantic unaries (Deeplab 3+ network) and affinities derived from Mask-RCNN foreground probability. Colors indicate predicted semantic classes with variations for separate instances. *Rightmost column:* Results for the sponge dataset. Cell-bodies are colored in blue, microvilli in green and flagella in red.

MRCNN[15]		GMIS[29]		Deeplab[8]		Cityscapes			
att	rep	att	rep	att	rep	sem	PQ	PQ Th	PQ St
✓	✓					✓	59.3	50.6	65.7
	✓	✓				✓	58.6	48.8	65.7
		✓	✓			✓	56.1	42.8	65.7
✓	✓	✓	✓	✓	✓	✓	48.7	38.7	55.9
		✓	✓	✓	✓	✓	47.3	35.5	55.9
				✓	✓	✓	46.3	33.1	56.0

Table 1: Panoptic segmentation quality PQ of the SMW on top of diverse sources of graph weights.

Cityscapes	PQ	PQ Th	PQ St
SMW	59.3	50.6	65.7
PFPN[18]	58.1	52.0	62.5
DIN[3]	53.8	42.5	62.1
Sponge	PQ	PQ Th	PQ St
SMW	51.6	62.1	20.0
MWS-MAX	48.1	56.2	23.8
CC _{sem}	43.4	55.6	06.7
CC _{aff}	24.3	27.7	13.9

Table 2: Comparison to other segmentation strategies.

in [table 2](#) and shown in [fig. 2](#) and the supplementary [fig. 3](#). Through observations on the images, we find that Mask-RCNN affinities are more reliable in detecting small objects as well as in connecting fragmented instances. Note that PQ mostly measures detection quality which is then weighted by the segmentation quality of the found instances, hence the detection strength of the Mask-RCNN shines through.

We observe that using all sources together leads to a performance drop of 10 percentage points below the best result. We believe this is due to the greedy nature of the SMW which selects the strongest of all provided edges. This example demonstrates how important it is to carefully select/train the algorithm input.

4.3. Semantic Instance Segmentation of 3D EM Volumes

Semantic instance segmentation is an important task in bio-medical image analysis where classes naturally arise through cellular structure. We use a 3D EM image dataset to compare the SMW to algorithms that separately optimize instance segmentation and semantic class assignment.

Dataset. The data-set consists of two FIBSEM volumes of a sponge choanocyte chamber. The data was acquired in [33] to investigate proto-neuronal cells in sponges using the segmentation approach introduced in [35]. These cells filter nutrients from water by creating a flow with the beating of a flagellum and absorbing the nutrients through microvilli that surround the flagellum in a collar [26] (see [fig. 2](#)). In order to investigate this process in detail, a precise semantic instance segmentation of the cell-bodies, flagella and microvilli is needed. The dataset consists of three EM image volumes of size $96 \times 896 \times 896$ pixel ($2 \times 18 \times 18 \mu\text{m}$).

Implementation Details. We predict affinities with two separate 3D U-Nets [9] to derive graph edge weights and semantic class probabilities respectively. We adopt the training procedure of [42] which uses the Dice Coefficient as the loss function. We use two volumes for training and one for testing.

We also implement baseline approaches which start from the same network predictions, but do not perform joint labeling and partitioning. First, we compare to instance segmentation with the Mutex Watershed, followed by assigning instances the semantic label of the strongest semantic edge

(MWS-MAX). In addition, we compute connected components of the semantic predictions (CC_{sem}) and short-range affinities (CC_{aff}).

Results. The PQ values in [table 2](#) show that the SMW outperforms the baselines approaches that separately optimize instance segmentation and semantic class assignment. An additional analysis can be found in the appendix [fig. 4](#), where we measure the runtimes for different volume sizes and observe almost linear scaling behavior.

5. Conclusion

We have introduced a new method for joint partitioning and labeling of weighted graphs as a generalization of the Mutex Watershed algorithm. We have shown that it optimally solves an objective function closely related to the objective of the Symmetric Multiway Cut problem. Our experiments demonstrate that SMW with graph edge weights predicted by convolutional neural networks outperform strong baselines on natural and biological images. Any improvement in the CNN performance will translate directly to an improvement of the SMW results. However, we also observe that the extreme value selection used by the SMW to assign edges to the active set can lead to sub-optimal performance when diverse edge weights sources are combined. Empirically, the algorithm scales almost linearly with the number of graph edges N making it applicable to large images and volumes without prior over-segmentation into superpixels. The source code will be made available upon publication.

References

- [1] Bjoern Andres, Kevin L. Briggman, Natalya Korogod, Graham Knott, Ullrich Koethe, and Fred A. Hamprecht. Globally Optimal Closed-Surface Segmentation for Connectomics. In *Computer Vision – ECCV 2012*, volume 7574, pages 778–791. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. [2](#)
- [2] Bjoern Andres, Jörg H. Kappes, Thorsten Beier, Ullrich Köthe, and Fred A. Hamprecht. Probabilistic image segmentation with closedness constraints. In *Computer Vision (ICCV), 2011 IEEE International Conference On*, pages 2611–2618. IEEE, 2011. [1, 2, 9](#)
- [3] Anurag Arnab and Philip HS Torr. Pixelwise instance segmentation with a dynamically instantiated network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 441–450, 2017. [6](#)
- [4] Min Bai and Raquel Urtasun. Deep watershed transform for instance segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2858–2866. IEEE, 2017. [2](#)
- [5] Thorsten Beier, Constantin Pape, Nasim Rahaman, Timo Prange, Stuart Berg, Davi D Bock, Albert Cardona, Graham W Knott, Stephen M Plaza, Louis K Scheffer, et al. Multicut brings automated neurite segmentation closer to human performance. *Nature Methods*, 14(2):101, 2017. [2](#)
- [6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. In *ICLR*, 2016. [2](#)
- [7] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. [2](#)
- [8] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. *arXiv preprint arXiv:1802.02611*, 2018. [5, 6, 10](#)
- [9] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3D U-Net: Learning dense volumetric segmentation from sparse annotation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 424–432. Springer, 2016. [6](#)
- [10] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. *arXiv:1604.01685 [cs]*, Apr. 2016. [5](#)
- [11] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009. [5](#)
- [12] Jifeng Dai, Kaiming He, and Jian Sun. Instance-Aware Semantic Segmentation via Multi-task Network Cascades. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3150–3158, Las Vegas, NV, USA, June 2016. IEEE. [2](#)
- [13] Alireza Fathi, Zbigniew Wojna, Vivek Rathod, Peng Wang, Hyun Oh Song, Sergio Guadarrama, and Kevin P. Murphy. Semantic Instance Segmentation via Deep Metric Learning. *arXiv:1703.10277 [cs]*, Mar. 2017. [2](#)
- [14] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59(2):167–181, Sept. 2004. [2](#)
- [15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. *arXiv:1703.06870 [cs]*, Mar. 2017. [2, 5, 6, 10](#)
- [16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*, Dec. 2015. [1](#)
- [17] Jörg H. Kappes, Markus Speth, Björn Andres, Gerhard Reinelt, and Christoph Schn. Globally optimal image partitioning by multcuts. In Yuri Boykov, Fredrik Kahl, Victor Lempitsky, and Frank R. Schmidt, editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 6819, pages 31–44. Springer Berlin Heidelberg, 2011. [2](#)
- [18] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic Feature Pyramid Networks. *arXiv:1901.02446 [cs]*, Jan. 2019. [2, 5, 6](#)

- [19] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic Segmentation. *arXiv:1801.00868 [cs]*, Jan. 2018. 1
- [20] Alexander Kirillov, Evgeny Levinkov, Bjoern Andres, Bogdan Savchynsky, and Carsten Rother. Instancecut: From edges to instances with multicut. In *CVPR*, volume 3, page 9, 2017. 2
- [21] Shu Kong and Charless Fowlkes. Recurrent Pixel Embedding for Instance Grouping. *arXiv preprint arXiv:1712.08273*, 2017. 2
- [22] N. E. Krasowski, T. Beier, G. W. Knott, U. Kothe, F. A. Hamprecht, and A. Kreshuk. Neuron Segmentation With High-Level Biological Priors. *IEEE Transactions on Medical Imaging*, 37(4):829–839, Apr. 2018. 2
- [23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, May 2017. 1
- [24] Thorben Kroeger, Jörg H. Kappes, Thorsten Beier, Ullrich Koethe, and Fred A. Hamprecht. Asymmetric Cuts: Joint Image Labeling and Partitioning. In *Pattern Recognition*, volume 8753, pages 199–211. Springer International Publishing, Cham, 2014. 1, 2, 3, 9
- [25] Joseph B Kruskal. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. *Proceedings of the American Mathematical Society*, page 3, 1956. 4
- [26] Paul-Friedrich Langenbruch and Norbert Weissenfels. Canal systems and choanocyte chambers in freshwater sponges (porifera, spongillidae). *Zoomorphology*, 107(1):11–16, 1987. 6
- [27] Kisuk Lee, Jonathan Zung, Peter Li, Viren Jain, and H Sebastian Seung. Superhuman Accuracy on the SNEMI3D Connectomics Challenge. *arXiv preprint arXiv:1706.00120*, 2017. 2, 5
- [28] Evgeny Levinkov, Jonas Uhrig, Siyu Tang, Mohamed Omran, Eldar Insafutdinov, Alexander Kirillov, Carsten Rother, Thomas Brox, Bernt Schiele, and Bjoern Andres. Joint Graph Decomposition & Node Labeling: Problem, Algorithms, Applications. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1904–1912, Honolulu, HI, July 2017. IEEE. 1, 2
- [29] Yiding Liu, Siyu Yang, Bin Li, Wengang Zhou, Ji-Zeng Xu, Houqiang Li, and Yan Lu. Affinity Derivation and Graph Merge for Instance Segmentation. In *The European Conference on Computer Vision (ECCV)*, page 18, Sept. 2018. 2, 5, 6, 10
- [30] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 2
- [31] Michael Maire, Takuya Narita, and Stella X. Yu. Affinity CNN: Learning pixel-centric pairwise relations for figure/ground embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 174–182, 2016. 2, 5
- [32] Francisco Massa and Ross Girshick. Maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. 2018. 5, 10
- [33] Jacob M Musser, Klaske J Schippers, Michael Nickel, Giulia Mizzon, Andrea B Kohn, Constantin Pape, Jörg U Hammel, Florian Wolf, Cong Liang, Ana Hernández-Plaza, et al. Profiling cellular diversity in sponges informs animal cell type and nervous system evolution. *BioRxiv*, page 758276, 2019. 6
- [34] Constantin Pape, Thorsten Beier, Peter Li, Viren Jain, Davi D. Bock, and Anna Kreshuk. Solving Large Multicut Problems for Connectomics via Domain Decomposition. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 1–10, Venice, Oct. 2017. IEEE. 2
- [35] Constantin Pape, Alex Matskevych, Adrian Wolny, Julian Hennies, Giulia Mizzon, Marion Louveaux, Jacob Musser, Alexis Maizel, Detlev Arendt, and Anna Kreshuk. Leveraging domain knowledge to improve microscopy image segmentation with lifted multcuts. *Frontiers in Computer Science*, 1:6, 2019. 6
- [36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv:1506.01497 [cs]*, June 2015. 2
- [37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015. 2
- [38] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs]*, Sept. 2014. 1
- [39] Joseph Tighe, Marc Niethammer, and Svetlana Lazebnik. Scene Parsing with Object Instance Inference Using Regions and Per-exemplar Detectors. *International Journal of Computer Vision*, 112(2):150–171, Apr. 2015. 1
- [40] Zhuowen Tu, Xiangrong Chen, Alan L. Yuille, and Song-Chun Zhu. Image Parsing: Unifying Segmentation, Detection, and Recognition. *International Journal of Computer Vision*, 63(2):113–140, July 2005. 1
- [41] Steffen Wolf, Alberto Bailoni, Constantin Pape, Nasim Rahaman, Anna Kreshuk, Ullrich Köthe, and Fred A Hamprecht. The Mutex Watershed and its Objective: Efficient, Parameter-Free Image Partitioning. *arXiv preprint arXiv:1904.12654*, May 2019. 1, 2, 4, 10
- [42] Steffen Wolf, Constantin Pape, Nasim Rahaman, Anna Kreshuk, Ullrich Köthe, and Fred Hamprecht. The Mutex Watershed: Efficient, Parameter-Free Image Partitioning. In *The European Conference on Computer Vision (ECCV)*, page 17, Sept. 2018. 1, 4, 5, 6
- [43] Jian Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 702–709, Providence, RI, June 2012. IEEE. 1
- [44] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Learning a Discriminative Fea-

- ture Network for Semantic Segmentation. *arXiv:1804.09337 [cs]*, Apr. 2018. [2](#)
- [45] Fisher Yu and Vladlen Koltun. Multi-Scale Context Aggregation by Dilated Convolutions. *arXiv:1511.07122 [cs]*, Nov. 2015. [2](#)
- [46] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890, 2017. [2](#), [5](#)

A. Symmetric Multiway Cut in Related Literature

We will now relate the Symmetric Multiway Cut definition in [eqs. \(1\) to \(6\)](#) with the the objective given in [\[24\]](#). In contrast to this work Kroeger *et al.* [\[24\]](#) do not split the set of edges in to attractive and repulsive edges. Instead they model repulsion with negative weights and formulate the SMWC as the following constrained energy minimization/integer linear program (ILP):

$$\min_{y \in \{0,1\}^{|E'|}} \left(\sum_{e \in E^S} w_e(1 - y_e) + \sum_{e \in E} w_e y_e \right) \quad \text{s.t. } y \in \text{SMWC}_{G'} \quad (\text{A.18})$$

The variables $y_e \in \{0, 1\}$ are indicators for cuts in the graph, i.e. when $y_e = 1$ the edge e is cut, and SMWC'_G is the polytope of consistent solutions defined by linear constraints:

$$\sum_{e \in c \setminus \{e^-\}} y_e \geq y_{e^-} \quad \forall e^- \in c \quad \forall c \in \text{Cycles in } G \quad (\text{A.19})$$

$$\sum_{t \in T} y_{it} = |T| - 1 \quad \forall i \in V \quad (\text{A.20})$$

$$y_{uv} \geq y_{ut} - y_{vt} \quad \forall (u, v) \in E; t \in T \quad (\text{A.21})$$

$$y_{uv} \geq y_{vt} - y_{ut} \quad \forall (u, v) \in E; t \in T \quad (\text{A.22})$$

The cycle constraints [\(A.19\)](#) form the so called Multicut polytope [\[2\]](#); they forbid dangling edges thus all non-cut internal edges form clusters on the graph G . [Equation \(A.20\)](#) ensures that each internal node is connected to exactly one terminal node. Finally, [\(A.21\)](#) and [\(A.22\)](#) are cycle constraints on all cycles with one terminal node; they enforce that an internal edge is always cut when its incident nodes are connected to different terminal nodes. This ensures that the resulting partitioning and labeling is always consistent. Note that an edge between two nodes connected to the same terminal is allowed to be cut, so two instances of the same class may touch.

We will now transform the objective given in [eq. \(A.18\)](#) and introduce an additional parameter p . Instead of finding a small-weight set to cut from the graph, we try to find a large-weight set $A \subseteq E$ to keep in the graph.

First, we split the internal edges into repulsive ($E^- := \{e \in E \mid w_e < 0\}$) and attractive edges ($E^+ := \{e \in E \mid w_e \geq 0\}$) so the energy function becomes

$$\sum_{e \in E^S} w_e^p(1 - y_e) + \sum_{e \in E^+} w_e^p y_e - \sum_{e \in E^-} |w_e|^p y_e. \quad (\text{A.23})$$

For $p = 1$ the ILP corresponds to the Symmetric Multiway Cut. Subtracting the constant sum of all positive edge

weights, using $w_e \leq 0 \forall e \in E^S$ yields

$$-\sum_{e \in E^S} |w_e|^p (1 - y_e) - \sum_{e \in E^+} w_e^p (1 - y_e) - \sum_{e \in E^-} |w_e|^p y_e. \quad (\text{A.24})$$

Finally, by substituting

$$a_e = \begin{cases} y_e & \text{if } e \in E^- \\ 1 - y_e & \text{if } e \in E^+ \cup E^S \end{cases} \quad (\text{A.25})$$

we obtain the equivalent objective

$$\begin{aligned} \max_{a \in \{0,1\}^{|E'|}} & \sum_{e \in E'} |w_e|^p a_e \\ \text{s.t. } & a \in \text{SMWC}_{G'}^a. \end{aligned} \quad (\text{A.26})$$

Here, $\text{SMWC}_{G'}^a$ is the polytope formed by [eqs. \(2\)](#) to [\(6\)](#). Since all weights are positive in the SMW graph, the absolute value is omitted in [eq. \(1\)](#).

B. Mutex and Label Constraints

We will now formally derive that the constraints in [eqs. \(14\)](#) and [\(15\)](#) are necessary for the Symmetric Multiway Cut constraints [eqs. \(A.19\)](#) to [\(A.22\)](#). Wolf *et al.* [41] show that the [eq. \(14\)](#) is necessary for the multicut constraints [eq. \(A.19\)](#). Therefore, it is left to show that

$$\left. \begin{aligned} \sum_{t \in T} y_{ut} &= |T| - 1 \quad \forall u \in V \\ y_{uv} &\geq y_{ut} - y_{vt} \quad \forall (u, v) \in E; t \in T \\ y_{uv} &\geq y_{vt} - y_{ut} \quad \forall (u, v) \in E; t \in T \end{aligned} \right\} \Rightarrow \mathcal{P}(A) = \emptyset. \quad (\text{A.27})$$

The right-hand side is the label constraint which will be shown to be a subset of the constraints formed by [eqs. \(A.20\)](#) to [\(A.22\)](#) (here on the left).

First we show by contradiction and using [eq. \(A.20\)](#) that an internal node $v \in V$ can only be connected to a single terminal $t \in T$: Assume that there is a $t^* \neq t$ which is also connected to v ; then we have $y_{ut} = 0$ and $y_{ut^*} = 0$. Now rewrite [eq. \(A.20\)](#) and insert these two variables so that we get the contradiction

$$1 = \sum_{t' \in T} (1 - y_{ut'}) = \sum_{t' \in T \setminus \{t, t^*\}} (1 - y_{ut'}) + 1 + 1 \geq 2, \quad (\text{A.28})$$

We further show that two connected nodes u and v are always connected to the same terminal node t . Without lossing generality we assume u and v are connected and u and t are connected, i.e. $y_{uv} = 0$ and $y_{ut} = 0$. Then [eqs. \(A.21\)](#) and [\(A.22\)](#) give us

$$\left. \begin{aligned} 0 &\geq 0 - y_{vt} \\ 0 &\geq y_{vt} - 0 \end{aligned} \right\} \Rightarrow y_{vt} = 0 \quad (\text{A.29})$$

Finally, we can prove [eq. \(A.27\)](#): any path starting from t begins with an edge (t, v) to some node v ; all nodes connected to v (and v itself) are connected to t and no other terminal node. Therefore there can not be any path from t to another terminal node t' satisfying the label constraint $\mathcal{P}(A) = \emptyset$.

C. Additional Details of the Cityscapes Experiments

C.1. Implementation Details

We use the class probabilities from a Deeplab 3+ [8] as semantic edge weights. We use a trained model provided by Tensorflow. employ the Mask-RCNN [15] implementation provided by [32] and trained a model on Cityscapes following [15]’s training configuration. The graph weights are derived as explained above. We derive graph weights for different offsets: for attractive edges we use (1) 8-neighbourhood with distances of $\{1, 2, 4\}$ pixels, (2) random pairs inside each bounding box. For repulsive edges we sample 5 random pixel pairs for each mask and compute the soft IOU ([eq. \(17\)](#)). [29] trained a Deeplab 3+ to predict affinities for their graph-clustering algorithm. They kindly provided their trained models allowing us to use the same affinities. Since their clustering utilizes a threshold, we treat the threshold as the splitting point between attractive and repulsive edge weights; affinities below the threshold are inverted and scaled to $[0, 1]$. In addition to the model by GMIS that is trained on scaled bounding boxes, we train a Deeplab3+ for affinity predictions on the full images. Because [29] only tackle instance segmentation, their model does not predict affinities for stuff classes. We train the network with Sorense Dice Loss and the same stencil pattern as [29]. The training protocol follows the settings in [8], using a batch size of 12 and 70k training iterations. We do not employ any test time augmentations.

C.2. Additional images

D. Scaling Behavior



Figure 3: Further examples panoptic results on Cityscapes using using semantic unaries (Deeplab 3+ network) and affinities derived from Mask-RCNN foreground probability. Prediction errors are highlighted in green.

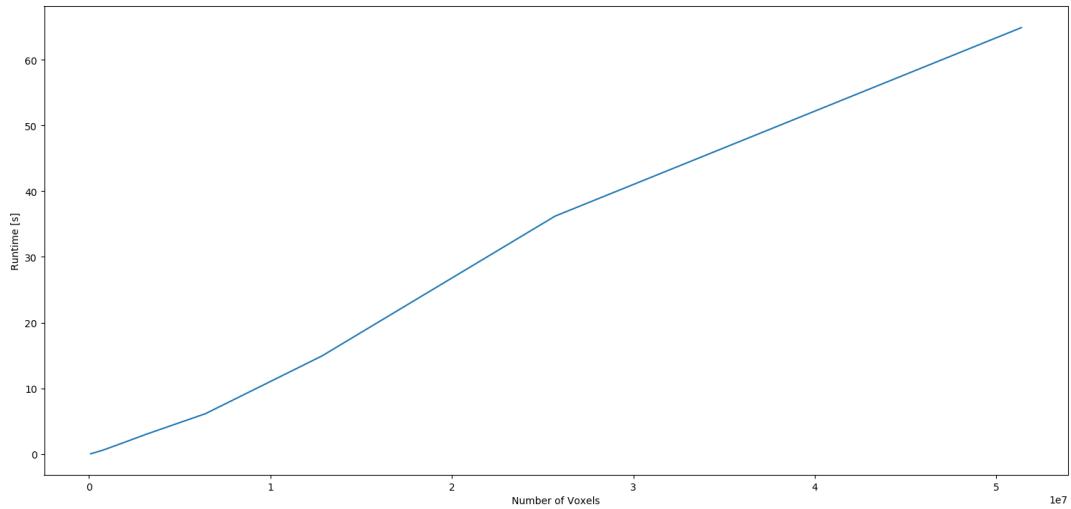


Figure 4: Runtime scaling of the SMW. We evaluate the runtime of the SMW for different volume sizes of the 3D Sponge dataset. We find an almost linear relation between runtime and number of voxels.