

000

# GASP, a generalized framework for 001 agglomerative clustering of signed graphs and its 002 application to Instance Segmentation

003

004

005 Anonymous ECCV submission

006

007 Paper ID 5039

008

009

010

011 **Abstract.** We propose a new theoretical framework that generalizes  
012 algorithms for hierarchical agglomerative clustering to weighted graphs  
013 with both attractive and repulsive interactions between the nodes. This  
014 framework defines GASP, a Generalized Algorithm for Signed graph Par-  
015 titioning, and allows us to explore many combinations of different linkage  
016 criteria and cannot-link constraints. We prove the equivalence of exist-  
017 ing clustering methods to some of those combinations, and introduce  
018 new algorithms for combinations which have not been studied before.  
019 On stochastic block model problems, GASP compares favorably to spec-  
020 tral clustering. More importantly, we conduct a systematic comparison  
021 of various instantiations of GASP in image instance segmentation prob-  
022 lems, in terms of accuracy but also efficiency and robustness to noise.  
023 We find that one of the new algorithms proposed in our framework out-  
024 performs all previously known agglomerative methods for signed graphs,  
025 both on the competitive CREMI 2016 EM segmentation benchmark and  
026 on the CityScapes dataset.

027

028

029 **Keywords:** Partitioning of Signed Graphs, Generalized Framework, Ag-  
030 glomerative Clustering, Instance Segmentation

031

## 1 Introduction

032 In computer vision, the partitioning of weighted graphs has been successfully  
033 applied to tasks as diverse as image segmentation, object tracking and pose  
034 estimation. Most graph clustering methods work with positive edge weights only,  
035 which can be interpreted as similarities or distances between the nodes. These  
036 methods require users to specify the desired numbers of clusters (as in spectral  
037 clustering) or a termination criterion (e.g. in iterated normalized cuts) or even  
038 to add a seed for each object (e.g. seeded watershed or random walker).

039 Other graph clustering methods work with so-called *signed graphs*, which  
040 include both positive and negative edge weights corresponding to attraction  
041 and repulsion between nodes. The advantage of signed graphs over positive-  
042 weighted graphs is that balancing attraction and repulsion allows us to obtain  
043 a clustering without defining additional parameters. A canonical formulation of  
044 the signed graph partitioning problem is the *multicut* or *correlation clustering*  
problem [35][12]. This problem is NP-hard, though many approximate solvers

have been proposed [49] [69] [7] [93]. The general problem of graph partitioning can also be solved approximately by greedy agglomerative clustering [39] [53] [89] [36]. Agglomerative clustering algorithms for signed graphs have clear advantages: they are parameter-free and efficient. Despite the fact that a variety of these algorithms exist, no overarching study has so far been conducted to compare their robustness and efficiency or to provide guidelines for matching an algorithm to the partitioning problem at hand.

In this paper, we propose a new theoretical framework that generalizes over agglomerative algorithms for signed graphs by linking them to hierarchical agglomerative clustering on positive-weighted graphs [48]. This framework defines an underlying basic algorithm and allows us to explore its combinations with different linkage criteria and *cannot-link constraints*. We then formally prove that some of the combinations correspond to existing clustering algorithms and introduce new algorithms for combinations which have not been explored before.

We evaluate and compare these algorithms on *instance segmentation* – a computer vision task of assigning each pixel of an image to an object instance. We use a CNN to predict the edge weights of a graph such that each node represents a pixel of the image, similarly to [62] [52] [89], and provide these weights as input to the algorithms in our framework (see Fig. 1).

With our comparison experiments, performed both on 2D urban scenes from the CityScapes dataset and 3D electron microscopy image volumes of neurons, we benchmark all algorithms in our framework, focusing on their efficiency, robustness and tendency to over- or under-cluster. We show that one of the new algorithms derived from our framework, based on an average linkage criterion, outperforms all previously known agglomeration methods expressed in the framework and that it achieves competitive performance on CityScapes and the challenging CREMI 2016 segmentation benchmark.

In Sec. 6, we also show how GASP outperforms spectral clustering methods on the task of neuron segmentation and how on synthetic graphs it achieves similar scores to a recently proposed spectral method for signed graphs.

## 2 Related work

**Proposal-based methods** have been highly successful in instance segmentation competitions like MS COCO [56], Pascal VOC2012 [22] and CityScapes [15]. They decompose the instance segmentation task into two steps that consists in generating object proposals and assigning to each bounding box a class and a binary segmentation mask [33] [73] [58] [92] [54] [47] [32] [9] [18] [55]. Other methods use instead recurrent models to sequentially generate instances one-by-one [77] [75].

**Proposal-free methods** adopt a bottom-up approach by directly grouping pixels into instances. Recently, there has been a growing interest for such methods that do not involve object detection, since, in certain types of data, object instances cannot be approximated by bounding boxes. For example, the approach proposed in [41] uses a combinatorial framework for instance segmentation, whereas a watershed transform is learned in [5] by also predicting its

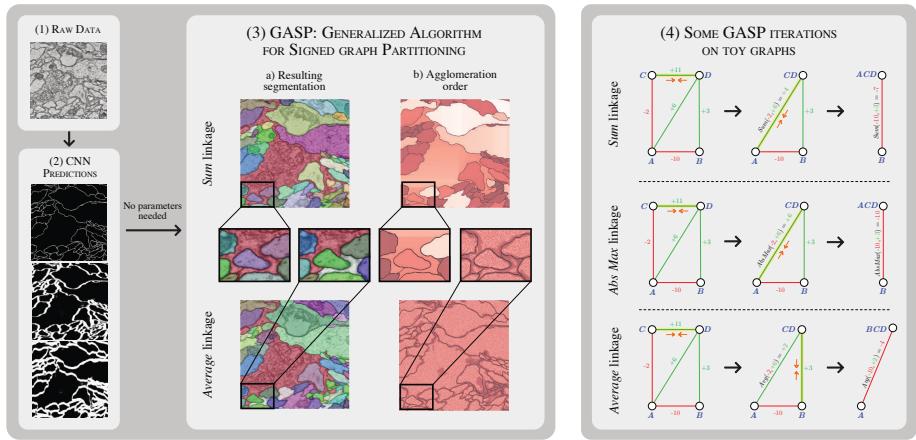


Fig. 1: GASP example. (1) Raw data from the CREMI 2016 neuron-segmentation challenge. (2) Some short- and long-range predictions of our CNN model, where white pixels represent boundary evidence. (3) Outputs of two agglomerative algorithms included in our proposed generalized clustering framework, with *Sum* and *Average* linkage criteria. The final clustering / instance segmentation is shown in 3a, overlaid with the raw image. The agglomeration order in 3b shows which pairs of neighboring pixels were merged first (white), later on (brown/red), or never (black). (4) Some iterations of GASP on toy graph examples with attractive/positive (green) and repulsive/negative (red) interactions. At each iteration, the yellow edge with highest interaction is contracted (orange arrows), until only negative edges are left in the graph.

gradient direction. Others use metric learning to predict high-dimensional associative pixel embeddings that map pixels of the same instance close to each other, while mapping pixels belonging to different instances further apart [51, 23, 67, 19]. Final instances are then retrieved by applying a clustering algorithm, like in the end-to-end trainable mean-shift pipeline of [44]. Other recent successful methods simply let the model predict the relative coordinates of the instance center [66, 10] or, given a point  $(x, y)$  in the image, they train a model to generate the mask of the instance located at  $(x, y)$  [83].

**Edge detection** also experienced recent progress thanks to deep learning, both on natural images [29, 62, 90, 43] and biological data [52, 82, 64, 14]. In neuron segmentation for connectomics, a field of neuroscience we also address in our experiments, boundaries are converted to final instances with subsequent postprocessing and superpixel-merging: some use loopy graphs [37, 45] or trees [64, 61, 59, 26, 86] to represent the region merging hierarchy; the lifted multicut [8] formulates the problem in a combinatorial framework, whereas flood-filling networks [34] and MaskExtend [64] use a CNN to iteratively grow one region/neuron at the time; recently, the work of [65] made the process more efficient by em-

ploying a combinatorial encoding of the segmentation. A structured learning approach was also proposed in [28][85].

**Agglomerative graph clustering** has often been applied to instance segmentation [3][76][60][80], because of its efficiency as compared to other top-down approaches like graph cuts. Novel termination criteria and merging strategies have often been proposed: the agglomeration in [63] deploys fixed sets of merge constraints; the popular graph-based method [24] stops the agglomeration when the merge costs exceed a measure of quality for the current clusters. The optimization approach in [40] performs greedy merge decisions that minimize a certain energy, while other pipelines use classical linkage criteria, e.g. average linkage [62][52], median [28] or a linkage learned by a random forest classifier [68][42].

**Clustering of signed graphs** has the goal of partitioning a graph with both attractive and repulsive cues. Finding an optimally balanced partitioning has a long history in combinatorial optimization [30][31][13]. NP-hardness of the *correlation clustering* problem was shown in [6], while the connection with graph multicut was made by [20]. Modern integer linear programming solvers can tackle problems of considerable size [2], but accurate approximations [69][7][93], greedy agglomerative algorithms [53][88][39][36] and persistence criteria [50][49] have been proposed for even larger graphs. Another line of research is given by spectral clustering methods that, on the other hand, require the user to specify the number of clusters in advance. Recently, some of these methods have been generalized to graphs with signed weights [16][11][46], whereas others let the user specify must-link and cannot-link constraints between clusters [74][87][17].

This work reformulates the clustering algorithms of [53][89][39] in a generalized framework and adopt ideas from the proposal-free methods [62][89][52] to predict long-range relationships between pixels.

### 3 Generalized framework for agglomerative clustering of signed graphs

In this section, we first define notation and then introduce one of our main contributions: a signed graph partitioning algorithm (Sec. 3.2) that can be seen as a generalization of several existing and new clustering algorithms (Sec. 3.3).

#### 3.1 Notation and graph formalism

We consider an undirected simple edge-weighted graph  $\mathcal{G}(V, E, w^+, w^-)$  with both attractive and repulsive edge attributes. In computer vision applications, the nodes can represent either pixels, superpixels or voxels. We call the set  $\Pi$  a *clustering* or *partitioning* with  $K$  clusters if  $V = \cup_{S \in \Pi} S$ ,  $S \cap S' = \emptyset$  for different clusters  $S, S' \in \Pi$  and every cluster  $S \in \Pi$  induces a connected subgraph of  $\mathcal{G}$ . We also denote as  $S_u$  the cluster associated with node  $u$ . The weight function  $w^+ : E \rightarrow \mathbb{R}^+$  associates to every edge a positive scalar attribute  $w_e^+ \in \mathbb{R}^+$  representing a merge affinity or a similarity measure: the higher this number,

the higher the inclination of the two incident vertices to be assigned to the same cluster<sup>1</sup>. On the other hand,  $w^- : E \rightarrow \mathbb{R}^+$  associates to each edge a split tendency  $w_e^- \in \mathbb{R}^+$ : the higher this weight, the more the incident vertices would like to be in different clusters. Graphs of the type  $\mathcal{G}(V, E, w^+, w^-)$  are also often defined as *signed graphs*  $\mathcal{G}(V, E, w)$ , featuring positive and negative edge weights  $w_e \in \mathbb{R}$ . Following the theoretical considerations in [50], we define these signed weights as  $w_e = w_e^+ - w_e^-$ . Some approaches directly compute  $w_e$ , whereas others compute  $w_e^+$  and  $w_e^-$  separately. In this formalism, graphs with purely attractive interactions are a special case of  $\mathcal{G}(V, E, w)$  with  $w_e \geq 0, \forall e \in E$ .

**Inter-cluster interaction** We call two clusters  $S_u, S_v$  *adjacent* if there exists at least one edge  $e_{ts} \in E$  connecting a node  $t \in S_u$  to a node  $s \in S_v$ . In hierarchical agglomerative clustering, the interaction  $\mathcal{W}(S_u, S_v)$  between the two clusters is usually defined as a function  $\mathcal{W} : \Pi \times \Pi \rightarrow \mathbb{R}$ , named *linkage criterion*, depending on the weights of *all* edges connecting clusters  $S_u$  and  $S_v$ , i.e.  $(S_u \times S_v) \cap E$ . All the linkage criteria tested in this article are listed and defined in Table 1.

### 3.2 GASP: generalized algorithm for signed graph partitioning

In Algorithm 1, we provide simplified pseudo-code for the proposed GASP algorithm. GASP implements a bottom-up approach that starts by assigning each node to its own cluster and then iteratively merges pairs of adjacent clusters. The algorithm has two variants. The first one (option *addCannotLinkConstraints* is **False**) starts by merging clusters with the strongest attractive interaction and stops once the remaining clusters share only mutual repulsive interactions (see iterations on toy graphs in block 4 of Fig. 1). After each merging iteration, the interaction between the merged cluster and its neighbors is updated according to one of the linkage criteria  $\mathcal{W}(S_u, S_v)$  listed in Table 1.

In the second variant (option *addCannotLinkConstraints* is **True**), Algorithm 1 also introduces cannot-link constraints, which represent mutual exclusion relationships between pairs of nodes that cannot be associated with the same cluster in the final clustering. This variant selects the pair of clusters with the highest absolute interaction  $|\mathcal{W}(S_u, S_v)|$ , so that the most attractive and the most repulsive pairs are analyzed first (see example in Fig. 2(b)). If the interaction is repulsive, then the two clusters are constrained and its members can never merge in subsequent steps. If the interaction is attractive, then the clusters are merged, provided that they were not previously constrained. The algorithm terminates when all the remaining clusters are constrained.

In Appendix 8.1, we comment on the algorithm's computational complexity  $\mathcal{O}(N^2 \log N)$  and present our implementation given by the edge contraction Algorithm 2 based on a *disjoint set data structure* and a *priority queue*.

---

<sup>1</sup> Note that other formalisms for positively weighted graphs associate distances to the edges, thus, the *lower* the edge weight, the higher the attraction between the two linked nodes, contrary to our definition of  $w^+$ .

---

**Algorithm 1** GASP: generalized algorithm for signed graph partitioning

---

**Input:** Graph  $\mathcal{G}(V, E, w^+, w^-)$ ; linkage criterion  $\mathcal{W}$ ; boolean `addCannotLinkConstraints`

**Output:** Final clustering  $\Pi$

- 1: Initial clustering:  $\Pi = \{\{v_1\}, \dots, \{v_{|V|}\}\}$
- 2: Initialize interactions between clusters with  $w_e = w_e^+ - w_e^-$
- 3: **repeat**
- 4:   Get  $S_u, S_v \in \Pi$  with highest interaction  $|\mathcal{W}(S_u, S_v)|$
- 5:   **if**  $[\mathcal{W}(S_u, S_v) > 0]$  **and**  $[S_u, S_v \text{ not constrained}]$  **then**
- 6:     Merge cluster  $S_u$  with  $S_v$
- 7:     Update interactions & constraints with neighboring clusters
- 8:   **else if** `addCannotLinkConstr` **and**  $[\mathcal{W}(S_u, S_v) \leq 0]$  **then**
- 9:     Add CannotLink Constraint between  $S_u$  and  $S_v$
- 10: **until** [all interactions between clusters are repulsive] **or**  
         [all adjacent clusters have cannot-link constraints]
- 11: **return**  $\Pi$

---

### 3.3 GASP with different linkage criteria: new and existing algorithms

Our main contribution is the generalized algorithm for signed graph partitioning, GASP, that encompasses several known and new agglomerative algorithms on display in Table 1. In our framework, individual algorithms are differentiated by the linkage criterion employed. We review them in the following paragraphs.

In the special case of an unsigned graph with only positive interactions, i.e.  $w_e^- = 0$  and  $w_e \geq 0 \forall e \in E$ , the algorithm performs a standard agglomerative hierarchical clustering by returning only a single cluster and a hierarchy of clusters defined by the order in which the clusters are merged (see Table 1, unsigned graphs).

Given a graph with both attractive and repulsive cues, an edge contraction algorithm with a sum update rule was pioneered in [53][39] (Table 1, *Sum* linkage). The authors present both a version with cannot-link constraints and one without, and then compare them with other greedy local-search algorithms approximating the multicut optimization problem. The Mutex Watershed [89] is another signed graph partitioning algorithm that introduces dynamical cannot-link constraints. In Proposition 81 (see Appendix 8.2) we prove that, surprisingly, it can also be seen as an efficient implementation of GASP with *Absolute maximum* linkage (def. in Table 1). Moreover, in Proposition 82 we also prove that GASP with *Abs Max* linkage returns the same clustering with or without enforcing cannot-link constraints. On the other hand, to our knowledge, *Average*, *Max* or *Min* linkage criteria have never been used for signed graph agglomerative algorithms or been combined with cannot-link constraints.

Apart from the linkage criteria defined in Table 1, additional ones were proposed in the literature: [68] for example uses a learned approach where a ran-

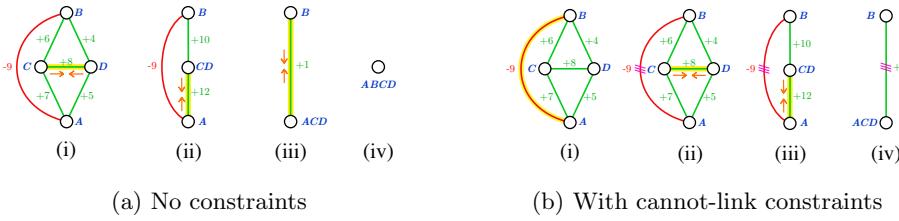


Fig. 2: Some iterations of the generalized algorithm (using *Sum* linkage criteria) with and without adding cannot-link constraints. The graph has both attractive (green) and repulsive (red) edges and cannot-link constraints are shown with triple violet bars on the edges. The edge selected at each iteration is highlighted in yellow. We note that when constraints are enforced, the final clustering is given by two clusters instead of only one.

dom forest classifier updates the cluster interactions depending on predefined edge and node features; other approaches introduce a weight regularization depending on the size of the clusters [24][36], whereas [28] uses a *quantile* linkage criterion by populating a histogram for each inter-cluster interaction. In our experiments, we decided to focus on the linkage criteria listed in Table 1, since they represent the most common options.

## 4 Experiments on neuron segmentation

We first evaluate and compare the agglomerative clustering algorithms described in the generalized framework on the task of neuron segmentation in electron microscopy (EM) image volumes. This application is of key interest in connectomics, a field of neuro-science with the goal of reconstructing neural wiring diagrams spanning complete central nervous systems. Currently, only proof-reading or manual tracing yields sufficient accuracy for correct circuit reconstruction [81], thus further progress is required in automated reconstruction methods.

EM segmentation is commonly performed by first predicting boundary pixels [8][14] or undirected affinities [89][52][28], which represent how likely it is for a pair of pixels to belong to the same neuron segment. The affinities do not have to be limited to immediately adjacent pixels. Thus, similarly to [52], we train a CNN to predict both short- and long-range affinities and use them as edge weights of a 3D grid graph, where each node represents a pixel/voxel of the volume image.

### 4.1 Data: CREMI challenge

We evaluate all algorithms in the proposed framework on the competitive CREMI 2016 EM Segmentation Challenge [27] that is currently the neuron segmentation challenge with the largest amount of training data available. The dataset comes from serial section EM of *Drosophila* fruit-fly tissue and consists of 6 volumes of

315 316 317	GASP linkage criteria $\mathcal{W}(S_u, S_v)$	Unsigned Graphs	Signed Graphs		315 316 317
			No Constraints	With Constraints	
318 319	Sum: $\sum_{e \in E_{uv}} w_e$	Sum Linkage Hier. Aggl. Clust.	GAEC [30]	Greedy Fixation [53]	318 319
320 321	Abs. Max: $w_e$ with $e = \arg \max_{t \in E_{uv}}  w_t $	Single Linkage Hier. Aggl. Clust.	Mutex Watershed [89]	Mutex Watershed [89]	320 321
322 323	Average: $\sum_{e \in E_{uv}} w_e /  E_{uv} $	Average Linkage Hier. Aggl. Clust.	NEW	NEW	322 323
324 325	Max: $\max_{e \in E_{uv}} w_e$	Single Linkage Hier. Aggl. Clust.	NEW	NEW	324 325
326 327	Min: $\min_{e \in E_{uv}} w_e$	Complete Linkage Hier. Aggl. Clust.	NEW	NEW	326 327

328 Table 1: Existing and new clustering algorithms that can be reformulated as  
329 special cases of the proposed generalized algorithm for signed graph partitioning,  
330 GASP, given a linkage criterion, a type of graph (signed or unsigned) and the  
331 optional use of cannot-link constraints. The set  $E_{uv}$  is defined as the set of all  
332 edges connecting cluster  $S_u$  to cluster  $S_v$ , i.e.  $E_{uv} = (S_u \times S_{v \neq u}) \cap E$ .  
333

334  
335  
336 1250x1250x125 voxels at resolution 4x4x40nm, three of which come with pub-  
337 licly available training ground truth. The results submitted to the leaderboard  
338 are evaluated using the CREMI score, based on the Adapted Rand-Score (Rand-  
339 Score) and the Variation of Information Score [4]. In Appendix 8.4 we provide  
340 more details about the training of our CNN model, inspired by work of [52, 28].  
341

## 342 4.2 Results and discussion

343  
344 **Comparison of linkage criteria** Table 2 shows how the agglomerative algo-  
345 rithms derived from our framework compare to each other. For a simple baseline,  
346 we also include a segmentation produced by thresholding the affinity predictions  
347 (THRESH). GASP with *Average* linkage, representing one of the new algorithms  
348 derived from our generalized framework, significantly outperformed all other pre-  
349 viously proposed agglomerative methods like GAEC [30] (GASP Sum), Greedy  
350 Fixation [53] (GASP Sum + Constraints) or Mutex Watershed [89] (GASP Abs.  
351 Max.). The competitive performance of this simple parameter-free algorithm is  
352 also reported in Table 3, showing the current leader-board of the challenge: all  
353 entries, apart from GASP, employ superpixel-based post-processing pipelines,  
354 several of which rely on the lifted multicut formulation of [8] that uses several  
355 random forests to predict graph edge weights, relying not only on information  
356 derived from affinity maps but also raw data and superpixel shape information.  
357 Note that the test volumes contain several imaging artifacts that make segmen-  
358 tation particularly challenging and might profit from more robust edge statistics  
359 of super-pixel based approaches. On the other hand, the fact that our algorithm

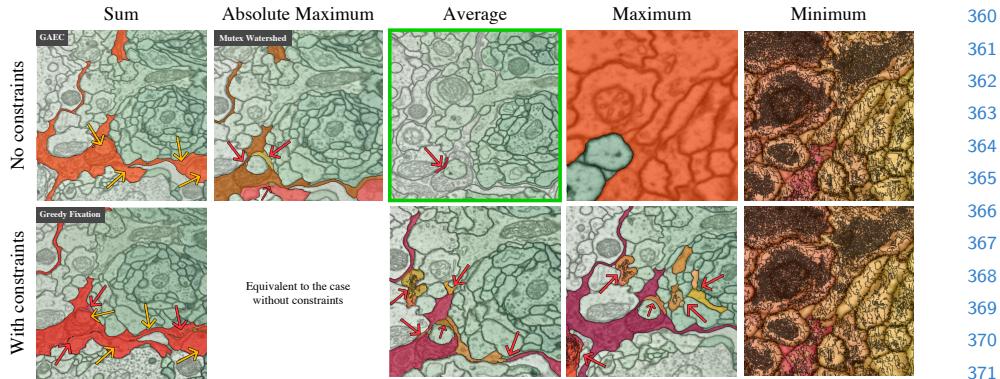


Fig. 3: Failure cases of GASP with different linkage criteria highlighted on some difficult parts of the CREMI Challenge data. Only the *wrongly* segmented regions are highlighted in different warm colors. Note that the data is 3D, hence the same color could be assigned to parts of segments that appear disconnected in 2D. Red arrows point to wrongly split regions. Yellow arrows point out merge errors. The *Average* linkage without cannot-link constraints returned the best segmentation.

can operate on pixels directly removes the parameter tuning necessary to obtain good super-pixels and can also avoid errors that result from wrong superpixels that cannot be fixed during later agglomeration. In Appendix 8.5 we provide more details about how we scaled up GASP to the full datasets. Appendix Table 7 lists the performances and the run-times for all tested GASP linkage criteria.

**Noise experiments** Additionally, we conduct a set of experiments where the CNN predictions are perturbed by structured noise, in order to highlight the properties of each GASP variant and perform an in-depth comparison that is as quantitative as possible. Appendix 8.6 introduces the type of spatially correlated noise that allowed us to perturb the CNN outputs by introducing simulated additional artifacts like missing or false positive boundary evidence. Fig. 4 summarizes our 12000 noise experiments: we focus on the best performing linkage criteria, i.e. *Average*, *Sum* and *Abs Max*, and test them with different amount of noise. In these experiments, we also want to assess how beneficial it is to use long-range CNN predictions in the agglomeration. Thus, we perform a set of simulations without adding long-range connections to the grid-graph and another set where we introduce them with a 10% probability<sup>2</sup>.

<sup>2</sup> We also performed experiments adding all the long-range predictions given by the CNN model, but we did not note major differences when using only 10% of them. Adding this fraction is usually sufficient to improve the scores.

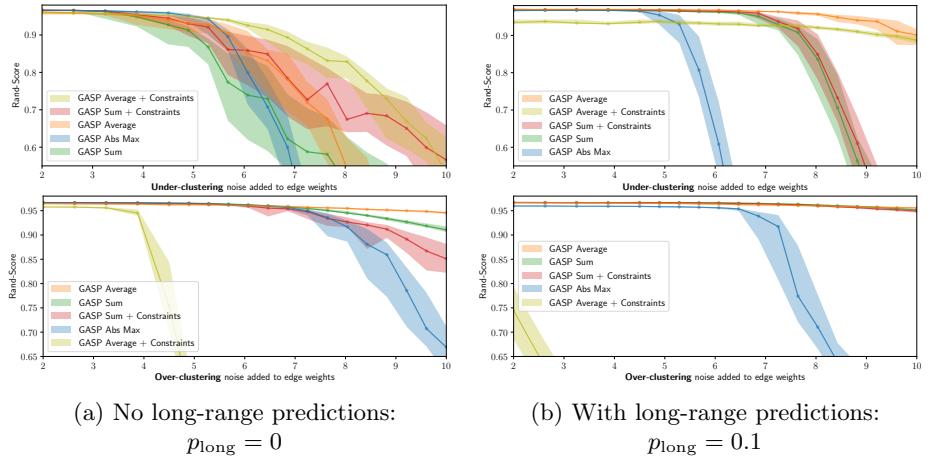


Fig. 4: GASP sensitivity to noise: *Average* linkage proved to be the most robust. Performances are given by Rand-Score (higher is better) depending on the amount of noise added to the CNN predictions. Solid lines represent median values over 30 experiments. Values between the 25th and the 75th percentile are shown in shaded areas. The two sets of experiments using under- and over-clustering noise are summarized in the plots at the top and at the bottom, respectively (see Appendix 8.6 for more details). For each experiment, some of the long-range CNN predictions were randomly selected with probability  $p_{\text{long}}$  and added as long-range edges to the pixel grid-graph. Experiments are performed on a crop of CREMI training sample B.

**Average and Abs Max linkage** Our findings confirm that GASP with *Average* linkage criterion represents the most robust algorithm tested and the one that benefits the most from using the long-range CNN predictions. On the other hand, it is not a surprise that the *Abs Max* statistic proposed by [89] is less robust to noise than the *Average* linkage, but, as we show in the Appendix Table 7, *Abs Max* represents a valid and considerably faster option. Adding long-range connections to the graph is generally helpful, but when many of them carry repulsive weights, then GASP with cannot-link constraints shows a clear tendency to over-cluster.

**Sum linkage** All our experiments show that GASP with *Sum* linkage is the algorithm with the highest tendency to under-cluster and incorrectly merge segments (see Fig. 3 for an example). This property is related to the empirical observation that a *Sum* statistic tends to grow clusters one after the other, as shown in Fig. 1 by the quite unique agglomeration order of the *Sum* statistic. An intuitive explanation of this fact is the following: initially, most of the intra-cluster nodes present similar attractive interactions between each others; when the two nodes sharing the most attractive interaction are merged, there is a high

Method	CREMI-Score (lower is better)
<b>GASP Average</b>	<b>0.226</b>
GASP Sum + Constraints [53]	0.282
GASP Abs. Max. [89]	0.322
GASP Max. + Constraints	0.324
GASP Sum [39]	0.334
GASP Average + Constraints	0.563
THRESH	1.521

Table 2: CREMI-Scores achieved by different linkage criteria and thresholding. All methods use the affinity predictions from our CNN as input. Scores are averaged over the three CREMI training datasets.

Method	CREMI-Score (lower is better)
Our CNN + DTWS + LMC	0.221
PNI CNN [52]	0.228
<b>Our CNN + GASP Average</b>	<b>0.241</b>
MALA CNN + MC [28]	0.276
CRU-Net [94]	0.566
LFC [70]	0.616

Table 3: Current leading entries in the CREMI challenge leaderboard [27] (March 2020). All entries, apart from ours using GASP, employ superpixel-based post-processing pipelines.

chance that they both share an attractive interaction with a common neighboring node, so the new interaction with this common neighbor will be immediately assigned to a high priority in the agglomeration, given by the sum of two high weights; this usually starts a “chain reaction”, where only a single cluster is agglomerated at the beginning. On the other hand, as we also see in Fig. 1, other linkage criteria like *Average* or *Abs Max* grow clusters of similar sizes in parallel and accumulate in this way much more reliable inter-cluster statistics.

## 5 Experiments on CityScapes

The segmentation performance of GASP is evaluated on the CityScapes dataset [15], which consists of 5000 street-scene images. Several of the current top instance segmentation methods on cityscapes predict affinities, e.g. SSAP [29] and GMIS [62]. Since [62] made the code and model publicly available, we used their pipeline consisting of two CNNs with similar architectures (one predicting semantic scores and the other pixel affinities between instances) and applied all the post-processing methods proposed by them, e.g. excluding background and resizing regions of interest. We then provided the output instance-affinities of the model as input to GASP. In [62] the instance-branch of the model was trained with a Binary Cross-Entropy loss, but we noticed how strong short-range boundary evidence was never predicted by the model. In Appendix 8.7 we present how we solved this problem by fine-tuning the model with *Sørensen-Dice* loss, similarly to [89]. Finally, the semantic categories were assigned to each instance by a majority vote based on the semantic output.

Results on the *test* set are summarized in Table 5. The best scores are achieved by Panoptic-DeepLab [10] that proposes a more powerful CNN model to regress the centers of the instances. The second best *proposal-free* method is SSAP [29], which predicts long-range instance-affinities similarly to GMIS [62], but trains the model with extra side-losses. GASP with *Average* linkage also achieves competitive results and outperforms the previous agglomeration method

Agglomeration method	AP
<b>GASP Average</b>	<b>34.3</b>
GASP Average + Constraints	33.9
MultiStepHAC [62]	33.0
GASP Abs. Max. [89]	32.1
GASP Sum + Constraints [53]	31.9
GASP Sum [39]	31.3

Table 4: Average Precision scores on CityScapes *val* achieved by the GMIS Model trained in [62] and different graph agglomeration methods.

Method	AP	AP 50% (higher is better)	495
Panoptic-DeepLab [10]	34.6	57.3	496
UPSNNet [91] †	33.0	59.6	497
SSAP [29]	32.7	51.8	498
AdaptIS [83]	32.5	52.5	499
PANet [58] †	31.8	57.1	500
<b>GMIS [62] + GASP Average</b>	<b>28.3</b>	<b>47.0</b>	501
JOSECB [66]	27.7	50.9	502
<b>GMIS [62]</b>	<b>27.3</b>	<b>45.6</b>	503
Mask R-CNN [33] †	26.2	49.9	504
SGN [57]	25.0	44.9	505

Table 5: Results on CityScapes test. Methods marked with † are *proposal-based*. Only methods that do not use external training data (such as MS COCO) are shown.

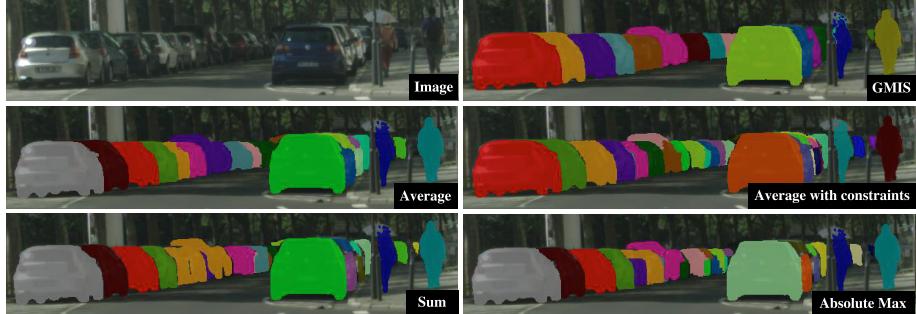


Fig. 5: Visual results given by different GASP linkage criteria on a crop of a CityScapes image from the *validation* set.

proposed in GMIS [62]. Similarly to the experiments on neuron segmentation, results on the *val* set (see Table 4 and Fig. 5) show that other GASP linkage criteria tend to over-cluster, e.g. *Abs Max*, or under-cluster and merge instances, e.g. *Sum*. The graph-merging algorithm proposed by [62] (MultiStepHAC) requires the user to tune several threshold parameters and when we applied it to the affinities predicted by our fine-tuned model it achieved an AP score of 33.0 on the *val* set, which is worse than the original value 34.1 reported in [62]. This is probably due to the fact that MultiStepHAC was tailored to the output affinities of the original model. Table 8 in Appendix includes the scores of all other tested GASP algorithms.

## 6 Comparison with spectral clustering

In this section we compare GASP to spectral clustering (SC) methods for signed graphs both on synthetic graphs and real ones from neuron segmentation. These

Method	Rand-Score
<b>GASP Average</b>	<b>0.8966</b>
GASP Sum	0.8965
GASP Abs Max	0.8932
SPONGE <sub>sym</sub> [16]	0.5839
$L_{sym}$ [46]	0.1931
SPONGE [16]	0.0789
BNC [11]	0.0074

Table 6: GASP compared to spectral clustering methods on a small crop of the CREMI dataset (sample B).

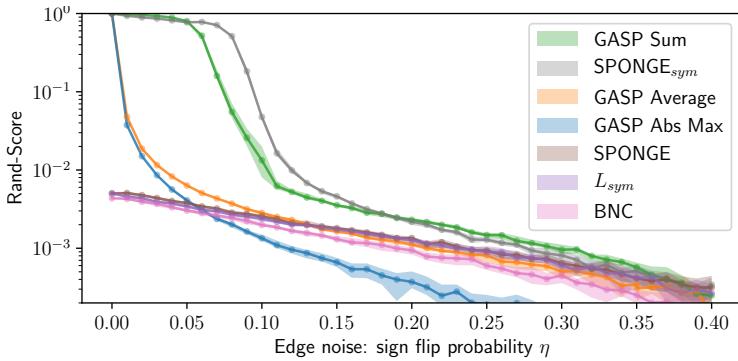


Fig. 6: GASP performances compared to spectral methods on synthetic graphs. The spectral methods were given the true number of clusters as input, in contrast to GASP.

methods require the user to specify the number of clusters in advance, in contrast to our proposed agglomeration method that determines the cluster number based on the signed weights of the graph. In the following comparison, we make these baselines as strong as possible by specifying the true number of clusters for the spectral methods.

**Synthetic graphs** – First, we compare the clustering performance on synthetic graphs generated by a signed stochastic block model (SSBM), where SC performs well. In particular, we used an Erdős-Renyi random graph model  $\mathcal{G}(N, p)$  with  $N = 10^5$  vertices and edge probability  $p = 0.1$ . Following the approach in [16], we partitioned the graph into  $k = 100$  equally-sized clusters, such that edges connecting vertices belonging to the same cluster (different clusters, respectively) had Gaussian distributed edge weights centered at  $\mu = 1$  ( $\mu = -1$ , respectively) and with standard deviation 0.1. To model noise, we flipped the sign of each edge independently with probability  $\eta \in [0, 0.4]$ . Median scores, 25th and 75th percentiles over 30 repetitions are shown in Fig. 6: GASP achieved scores comparable to SPONGE<sub>sym</sub> [16], a recently proposed SC method. The specific

585 design choice of a *sign flipping* noise used in the SSBM experiments turned out  
 586 to favor GASP with *Sum* linkage that, according to the experiments presented  
 587 in Sec. 4.2, is the one with the lowest tendency to over-cluster and grows one  
 588 cluster at the time.

589 **Neuron segmentation** – We extended the comparison with spectral meth-  
 590 ods to the task of neuron segmentation. Since SC cannot scale to the full CREMI  
 591 dataset, we evaluated GASP and SC on a smaller  $10 \times 100 \times 100$  voxels volume  
 592 resulting in a graph with  $10^5$  nodes and  $\sim 10^6$  edges. Scores are summarized  
 593 in Table 6. Despite the fact that the true number of ground truth clusters was  
 594 given as an input to the SC methods, GASP significantly outperformed them on  
 595 neuro-data. SC methods seem to have more difficulties when the graph is sparse.  
 596 Moreover, they did not handle well pixels on the boundaries between segments  
 597 and tended to cluster them together. Increasing  $k$  did not improve their scores  
 598 either.

## 600 7 Conclusion

601 We have presented a novel unifying framework for agglomerative clustering of  
 602 graphs with both positive and negative edge weights; and we have shown that  
 603 several existing clustering algorithms, e.g. the Mutex Watershed, can be reformu-  
 604 lated as special cases of one underlying agglomerative algorithm. This framework  
 605 also allowed us to introduce new algorithms, one of which, based on an *Aver-  
 606 age* linkage criterion, outperformed all the others: it proved to be a simple and  
 607 remarkably robust approach to process short- and long-range predictions of a  
 608 CNN applied to an instance segmentation task. On biological images, this sim-  
 609 ple average agglomeration algorithm can represent a valuable choice for a user  
 610 who is not willing to spend much time tuning complex task-dependent pipelines  
 611 based on superpixels.

## 613 614 References

1. Ailon, N., Charikar, M., Newman, A.: Aggregating inconsistent information: ranking and clustering. *Journal of the ACM (JACM)* **55**(5), 23 (2008)
2. Andres, B., Kroeger, T., Briggman, K.L., Denk, W., Korogod, N., Knott, G., Koethe, U., Hamprecht, F.A.: Globally optimal closed-surface segmentation for connectomics. In: European Conference on Computer Vision. pp. 778–791. Springer (2012)
3. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence* **33**(5), 898–916 (2011)
4. Arganda-Carreras, I., Turaga, S.C., Berger, D.R., Cireşan, D., Giusti, A., Gambardella, L.M., Schmidhuber, J., Laptev, D., Dwivedi, S., Buhmann, J.M., et al.: Crowdsourcing the creation of image segmentation algorithms for connectomics. *Frontiers in neuroanatomy* **9**, 142 (2015)
5. Bai, M., Urtasun, R.: Deep watershed transform for instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5221–5229 (2017)

- 630 6. Bansal, N., Blum, A., Chawla, S.: Correlation clustering. *Machine learning* **56**(1-3),  
631 89–113 (2004) 630  
632 7. Beier, T., Andres, B., Köthe, U., Hamprecht, F.A.: An efficient fusion move algo-  
633 rithm for the minimum cost lifted multicut problem. In: European Conference on  
634 Computer Vision. pp. 715–730. Springer (2016) 632  
635 8. Beier, T., Pape, C., Rahaman, N., Prange, T., Berg, S., Bock, D.D., Cardona, A.,  
636 Knott, G.W., Plaza, S.M., Scheffer, L.K., et al.: Multicut brings automated neurite  
637 segmentation closer to human performance. *Nature Methods* **14**(2), 101 (2017) 634  
638 9. Chen, Y.T., Liu, X., Yang, M.H.: Multi-instance object segmentation with occlu-  
639 sion handling. In: Proceedings of the IEEE Conference on Computer Vision and  
640 Pattern Recognition. pp. 3470–3478 (2015) 637  
641 10. Cheng, B., Collins, M.D., Zhu, Y., Liu, T., Huang, T.S., Adam, H., Chen, L.C.:  
642 Panoptic-DeepLab. arXiv preprint arXiv:1910.04751 (2019) 640  
643 11. Chiang, K.Y., Whang, J.J., Dhillon, I.S.: Scalable clustering of signed networks  
644 using balance normalized cut. In: Proceedings of the 21st ACM international con-  
645 ference on Information and knowledge management. pp. 615–624. ACM (2012) 642  
646 12. Chopra, S., Rao, M.R.: On the multiway cut polyhedron. *Networks* **21**(1), 51–89  
647 (1991) 644  
648 13. Chopra, S., Rao, M.R.: The partition problem. *Mathematical Programming* **59**(1-  
649 3), 87–115 (1993) 646  
650 14. Ciresan, D., Giusti, A., Gambardella, L.M., Schmidhuber, J.: Deep neural networks  
651 segment neuronal membranes in electron microscopy images. In: Advances in neural  
652 information processing systems. pp. 2843–2851 (2012) 648  
653 15. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R.,  
654 Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene  
655 understanding. In: Proceedings of the IEEE conference on computer vision and  
656 pattern recognition. pp. 3213–3223 (2016) 651  
657 16. Cucuringu, M., Davies, P., Glielmo, A., Tyagi, H.: SPONGE: A generalized eigen-  
658 problem for clustering signed networks. In: AISTATS (2019) 655  
659 17. Cucuringu, M., Koutis, I., Chawla, S., Miller, G., Peng, R.: Simple and scalable  
660 constrained clustering: a generalized spectral method. In: Artificial Intelligence and  
661 Statistics. pp. 445–454 (2016) 656  
662 18. Dai, J., He, K., Sun, J.: Instance-aware semantic segmentation via multi-task net-  
663 work cascades. In: Proceedings of the IEEE Conference on Computer Vision and  
664 Pattern Recognition. pp. 3150–3158 (2016) 659  
665 19. De Brabandere, B., Neven, D., Van Gool, L.: Semantic instance segmentation with  
666 a discriminative loss function. arXiv preprint arXiv:1708.02551 (2017) 662  
667 20. Demaine, E.D., Emanuel, D., Fiat, A., Immorlica, N.: Correlation clustering in  
668 general weighted graphs. *Theoretical Computer Science* **361**(2-3), 172–187 (2006) 663  
669 21. Dice, L.R.: Measures of the amount of ecologic association between species. *Ecology*  
670 **26**(3), 297–302 (1945) 666  
671 22. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The Pascal  
672 Visual Object Classes (VOC) challenge. *International journal of computer vision*  
673 **88**(2), 303–338 (2010) 667  
674 23. Fathi, A., Wojna, Z., Rathod, V., Wang, P., Song, H.O., Guadarrama, S., Murphy,  
675 K.P.: Semantic instance segmentation via deep metric learning. arXiv preprint  
676 arXiv:1703.10277 (2017) 671  
677 24. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation.  
678 *International journal of computer vision* **59**(2), 167–181 (2004) 673  
679

- 675 25. Finkel, J.R., Manning, C.D.: Enforcing transitivity in coreference resolution. In:  
676 Proceedings of the 46th Annual Meeting of the Association for Computational Lin-  
677 guistics on Human Language Technologies: Short Papers. pp. 45–48. Association  
678 for Computational Linguistics (2008)
- 679 26. Funke, J., Hamprecht, F.A., Zhang, C.: Learning to segment: training hierarchical  
680 segmentation under a topological loss. In: International Conference on Medical Im-  
681 age Computing and Computer-Assisted Intervention. pp. 268–275. Springer (2015)
- 682 27. Funke, J., Saalfeld, S., Bock, D., Turaga, S., Perlman, E.: Cremi challenge. [https://](https://cremi.org/)  
683 [cremi.org.](https://cremi.org/) (2016), accessed: 2019-11-15
- 684 28. Funke, J., Tschopp, F.D., Grisaitis, W., Sheridan, A., Singh, C., Saalfeld, S.,  
685 Turaga, S.C.: Large scale image segmentation with structured loss based deep  
686 learning for connectome reconstruction. IEEE transactions on pattern analysis and  
687 machine intelligence (2018)
- 688 29. Gao, N., Shan, Y., Wang, Y., Zhao, X., Yu, Y., Yang, M., Huang, K.: SSAP:  
689 Single-shot instance segmentation with affinity pyramid. In: The IEEE Interna-  
690 tional Conference on Computer Vision (ICCV) (October 2019)
- 691 30. Grötschel, M., Wakabayashi, Y.: A cutting plane algorithm for a clustering prob-  
692 lem. Mathematical Programming **45**(1-3), 59–96 (1989)
- 693 31. Grötschel, M., Wakabayashi, Y.: Facets of the clique partitioning polytope. Math-  
694 ematical Programming **47**(1-3), 367–387 (1990)
- 695 32. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Simultaneous detection and  
696 segmentation. In: European Conference on Computer Vision. pp. 297–312. Springer  
697 (2014)
- 698 33. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the  
699 IEEE international conference on computer vision. pp. 2961–2969 (2017)
- 700 34. Januszewski, M., Kornfeld, J., Li, P.H., Pope, A., Blakely, T., Lindsey, L., Maitin-  
701 Shepard, J., Tyka, M., Denk, W., Jain, V.: High-precision automated reconstruc-  
702 tion of neurons with flood-filling networks. Nature methods **15**(8), 605 (2018)
- 703 35. Kappes, J.H., Speth, M., Andres, B., Reinelt, G., Schnörr, C.: Globally optimal  
704 image partitioning by multicuts. In: International Workshop on Energy Minimiza-  
705 tion Methods in Computer Vision and Pattern Recognition. pp. 31–44. Springer  
706 (2011)
- 707 36. Kardoost, A., Keuper, M.: Solving minimum cost lifted multicut problems by node  
708 agglomeration. In: ACCV 2018, 14th Asian Conference on Computer Vision. Perth,  
709 Australia (2018)
- 710 37. Kaynig, V., Vazquez-Reina, A., Knowles-Barley, S., Roberts, M., Jones, T.R.,  
711 Kasthuri, N., Miller, E., Lichtman, J., Pfister, H.: Large-scale automatic recon-  
712 struction of neuronal processes from electron microscopy images. Medical image  
713 analysis **22**(1), 77–88 (2015)
- 714 38. Kernighan, B.W., Lin, S.: An efficient heuristic procedure for partitioning graphs.  
715 Bell system technical journal **49**(2), 291–307 (1970)
- 716 39. Keuper, M., Levinkov, E., Bonneel, N., Lavoué, G., Brox, T., Andres, B.: Efficient  
717 decomposition of image and mesh graphs by lifted multicuts. In: Proceedings of  
718 the IEEE International Conference on Computer Vision. pp. 1751–1759 (2015)
- 719 40. Kiran, B.R., Serra, J.: Global-local optimizations by hierarchical cuts and climbing  
720 energies. Pattern Recognition **47**(1), 12–24 (2014)
- 721 41. Kirillov, A., Levinkov, E., Andres, B., Savchynskyy, B., Rother, C.: Instancecut:  
722 from edges to instances with multicut. In: Proceedings of the IEEE Conference on  
723 Computer Vision and Pattern Recognition. pp. 5008–5017 (2017)

- 720 42. Knowles-Barley, S., Kaynig, V., Jones, T.R., Wilson, A., Morgan, J., Lee, D.,  
721 Berger, D., Kasthuri, N., Lichtman, J.W., Pfister, H.: RhoanaNet pipeline: Dense  
722 automatic neural annotation. arXiv preprint arXiv:1611.06973 (2016)  
723 43. Kokkinos, I.: Pushing the boundaries of boundary detection using deep learning.  
arXiv preprint arXiv:1511.07386 (2015)  
724 44. Kong, S., Fowlkes, C.C.: Recurrent pixel embedding for instance grouping. In:  
Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.  
pp. 9018–9028 (2018)  
725 45. Krasowski, N., Beier, T., Knott, G.W., Koethe, U., Hamprecht, F.A., Kreshuk, A.:  
Improving 3D EM data segmentation by joint optimization over boundary evidence  
726 and biological priors. In: 2015 IEEE 12th International Symposium on Biomedical  
727 Imaging (ISBI). pp. 536–539. IEEE (2015)  
728 46. Kunegis, J., Schmidt, S., Lommatzsch, A., Lerner, J., De Luca, E.W., Albayrak,  
S.: Spectral analysis of signed graphs for clustering, prediction and visualization.  
SIAM (2010)  
729 47. Ladický, L., Sturges, P., Alahari, K., Russell, C., Torr, P.H.: What, where and  
how many? combining object detectors and CRFS. In: European conference on  
730 computer vision. pp. 424–437. Springer (2010)  
731 48. Lance, G.N., Williams, W.T.: A general theory of classificatory sorting strategies:  
1. Hierarchical systems. The computer journal **9**(4), 373–380 (1967)  
732 49. Lange, J.H., Andres, B., Swoboda, P.: Combinatorial persistency criteria for multi-cut  
and max-cut. In: Proceedings of the IEEE Conference on Computer Vision  
733 and Pattern Recognition. pp. 6093–6102 (2019)  
734 50. Lange, J.H., Karrenbauer, A., Andres, B.: Partial optimality and fast lower bounds  
for weighted correlation clustering. In: International Conference on Machine Learning.  
pp. 2898–2907 (2018)  
735 51. Lee, K., Lu, R., Luther, K., Seung, H.S.: Learning dense voxel embeddings for 3D  
neuron reconstruction. arXiv preprint arXiv:1909.09872 (2019)  
736 52. Lee, K., Zung, J., Li, P., Jain, V., Seung, H.S.: Superhuman accuracy on the  
SNEMI3D connectomics challenge. arXiv preprint arXiv:1706.00120 (2017)  
737 53. Levinkov, E., Kirillov, A., Andres, B.: A comparative study of local search algo-  
rithms for correlation clustering. In: German Conference on Pattern Recognition.  
pp. 103–114. Springer (2017)  
738 54. Li, Y., Qi, H., Dai, J., Ji, X., Wei, Y.: Fully convolutional instance-aware semantic  
segmentation. In: Proceedings of the IEEE Conference on Computer Vision and  
739 Pattern Recognition. pp. 2359–2367 (2017)  
740 55. Liang, X., Wei, Y., Shen, X., Jie, Z., Feng, J., Lin, L., Yan, S.: Reversible recursive  
741 instance-level object segmentation. In: Proceedings of the IEEE Conference on  
Computer Vision and Pattern Recognition. pp. 633–641 (2016)  
742 56. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P.,  
Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference  
on computer vision. pp. 740–755. Springer (2014)  
743 57. Liu, S., Jia, J., Fidler, S., Urtasun, R.: Sgn: Sequential grouping networks for  
instance segmentation. In: Proceedings of the IEEE International Conference on  
Computer Vision. pp. 3496–3504 (2017)  
744 58. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance  
segmentation. In: Proceedings of the IEEE Conference on Computer Vision and  
745 Pattern Recognition. pp. 8759–8768 (2018)  
746 59. Liu, T., Jones, C., Seyedhosseini, M., Tasdizen, T.: A modular hierarchical ap-  
proach to 3D electron microscopy image segmentation. Journal of neuroscience  
methods **226**, 88–102 (2014)

- 765 60. Liu, T., Seyedhosseini, M., Tasdizen, T.: Image segmentation using hierarchical  
766 merge tree. *IEEE transactions on image processing* **25**(10), 4596–4607 (2016) 765  
767 61. Liu, T., Zhang, M., Javanmardi, M., Ramesh, N., Tasdizen, T.: SSHMT: Semi-  
768 supervised hierarchical merge tree for electron microscopy image segmentation. In:  
769 European Conference on Computer Vision. pp. 144–159. Springer (2016) 766  
770 62. Liu, Y., Yang, S., Li, B., Zhou, W., Xu, J., Li, H., Lu, Y.: Affinity derivation and  
771 graph merge for instance segmentation. In: Proceedings of the European Conference  
772 on Computer Vision (ECCV). pp. 686–703 (2018) 767  
773 63. Malmberg, F., Strand, R., Nyström, I.: Generalized hard constraints for graph  
774 segmentation. In: Scandinavian Conference on Image Analysis. pp. 36–47. Springer  
775 (2011) 772  
776 64. Meirovitch, Y., Matveev, A., Saribekyan, H., Budden, D., Rolnick, D., Odor, G.,  
777 Knowles-Barley, S., Jones, T.R., Pfister, H., Lichtman, J.W., et al.: A multi-pass  
778 approach to large-scale connectomics. arXiv preprint arXiv:1612.02120 (2016) 775  
779 65. Meirovitch, Y., Mi, L., Saribekyan, H., Matveev, A., Rolnick, D., Shavit, N.: Cross-  
780 classification clustering: An efficient multi-object tracking technique for 3-D in-  
781 stance segmentation in connectomics. In: Proceedings of the IEEE Conference on  
782 Computer Vision and Pattern Recognition. pp. 8425–8435 (2019) 777  
783 66. Neven, D., Brabandere, B.D., Proesmans, M., Gool, L.V.: Instance segmentation by  
784 jointly optimizing spatial embeddings and clustering bandwidth. In: Proceedings of  
785 the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8837–8845  
786 (2019) 782  
787 67. Newell, A., Huang, Z., Deng, J.: Associative embedding: End-to-end learning for  
788 joint detection and grouping. In: Advances in Neural Information Processing Systems.  
789 pp. 2277–2287 (2017) 785  
790 68. Nunez-Iglesias, J., Kennedy, R., Parag, T., Shi, J., Chklovskii, D.B.: Machine learn-  
791 ing of hierarchical clustering to segment 2D and 3D images. *PLoS one* **8**(8), e71715  
792 (2013) 788  
793 69. Pape, C., Beier, T., Li, P., Jain, V., Bock, D.D., Kreshuk, A.: Solving large multicut  
794 problems for connectomics via domain decomposition. In: Proceedings of the IEEE  
795 International Conference on Computer Vision. pp. 1–10 (2017) 791  
796 70. Parag, T., Tschopp, F., Grisaitis, W., Turaga, S.C., Zhang, X., Matejek, B., Ka-  
797 mentsky, L., Lichtman, J.W., Pfister, H.: Anisotropic EM segmentation by 3D  
798 affinity learning and agglomeration. arXiv preprint arXiv:1707.08935 (2017) 793  
799 71. Perlin, K.: An image synthesizer. *ACM Siggraph Computer Graphics* **19**(3), 287–  
800 296 (1985) 795  
801 72. Perlin, K.: Noise hardware. Real-Time Shading SIGGRAPH Course Notes (2001) 797  
802 73. Porzi, L., Bulo, S.R., Colovic, A., Kontschieder, P.: Seamless scene segmentation.  
803 In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recog-  
804 nition. pp. 8277–8286 (2019) 798  
805 74. Rangapuram, S.S., Hein, M.: Constrained 1-spectral clustering. In: AISTATS.  
806 vol. 30, p. 90 (2012) 801  
807 75. Ren, M., Zemel, R.S.: End-to-end instance segmentation with recurrent attention.  
808 In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recog-  
809 nition. pp. 6656–6664 (2017) 804  
810 76. Ren, Z., Shakhnarovich, G.: Image segmentation by cascaded region agglomera-  
811 tion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern  
812 Recognition. pp. 2011–2018 (2013) 805  
813 77. Romera-Paredes, B., Torr, P.H.S.: Recurrent instance segmentation. In: European  
814 conference on computer vision. pp. 312–329. Springer (2016) 806  
815

- 810 78. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical  
811 image segmentation. In: International Conference on Medical image computing and  
812 computer-assisted intervention. pp. 234–241. Springer (2015)
- 813 79. Saalfeld, S., Fetter, R., Cardona, A., Tomancak, P.: Elastic volume reconstruction  
814 from series of ultra-thin microscopy sections. *Nature methods* **9**(7), 717 (2012)
- 815 80. Salembier, P., Garrido, L.: Binary partition tree as an efficient representation for  
816 image processing, segmentation, and information retrieval. *IEEE transactions on Image Processing* **9**(4), 561–576 (2000)
- 817 81. Schlegel, P., Costa, M., Jefferis, G.S.: Learning from connectomics on the fly. *Current opinion in insect science* **24**, 96–105 (2017)
- 818 82. Schmidt, U., Weigert, M., Broaddus, C., Myers, G.: Cell detection with star-  
819 convex polygons. In: International Conference on Medical Image Computing and  
820 Computer-Assisted Intervention. pp. 265–273. Springer (2018)
- 821 83. Sofiuk, K., Baranova, O., Konushin, A.: AdaptIS: Adaptive instance selection net-  
822 work. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 7355–7363 (2019)
- 823 84. Sørensen, T.: A method of establishing groups of equal amplitude in plant sociology  
824 based on similarity of species and its application to analyses of the vegetation on  
825 danish commons. *Biol. Skr.* **5**, 1–34 (1948)
- 826 85. Turaga, S.C., Briggman, K.L., Helmstaedter, M., Denk, W., Seung, H.S.: Maximin  
827 affinity learning of image segmentation pp. 1865–1873 (2009)
- 828 86. Uzunbas, M.G., Chen, C., Metaxas, D.: An efficient conditional random field ap-  
829 proach for automatic and interactive neuron segmentation. *Medical image analysis* **27**, 31–44 (2016)
- 830 87. Wang, X., Qian, B., Davidson, I.: On constrained spectral clustering and its appli-  
831 cations. *Data Mining and Knowledge Discovery* **28**(1), 1–30 (2014)
- 832 88. Wolf, S., Bailoni, A., Pape, C., Rahaman, N., Kreshuk, A., Köthe, U., Hamprecht,  
833 F.A.: The mutex watershed and its objective: Efficient, parameter-free image par-  
834 titioning. *arXiv preprint arXiv:1904.12654* (2019)
- 835 89. Wolf, S., Pape, C., Bailoni, A., Rahaman, N., Kreshuk, A., Kothe, U., Hamprecht,  
836 F.: The mutex watershed: Efficient, parameter-free image partitioning. In: Pro-  
837 ceedings of the European Conference on Computer Vision (ECCV). pp. 546–562  
838 (2018)
- 839 90. Xie, S., Tu, Z.: Holistically-nested edge detection. In: Proc. ICCV'15. pp. 1395–  
840 1403 (2015)
- 841 91. Xiong, Y., Liao, R., Zhao, H., Hu, R., Bai, M., Yumer, E., Urtasun, R.: Upsnet:  
842 A unified panoptic segmentation network. In: Proceedings of the IEEE Conference  
843 on Computer Vision and Pattern Recognition. pp. 8818–8826 (2019)
- 844 92. Yang, Y., Hallman, S., Ramanan, D., Fowlkes, C.C.: Layered object models for  
845 image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelli-  
846 gence* **34**(9), 1731–1743 (2012)
- 847 93. Yarkony, J., Ihler, A., Fowlkes, C.C.: Fast planar correlation clustering for image  
848 segmentation. In: European Conference on Computer Vision. pp. 568–581. Springer  
849 (2012)
- 850 94. Zeng, T., Wu, B., Ji, S.: DeepEM3D: approaching human-level performance on 3D  
851 anisotropic EM image segmentation. *Bioinformatics* **33**(16), 2555–2562 (2017)