An Efficient Fusion Move Algorithm for the Minimum Cost Lifted Multicut Problem

Thorsten Beier[1(✉)], Björn Andres[2], Ullrich Köthe[1], and Fred A. Hamprecht[1]

[1] HCI/IWR, University of Heidelberg, Heidelberg, Germany
{thorsten.beier,ullrich.koethe,fred.hamprecht}@iwr.uni-heidelberg.de
[2] Computer Vision and Multimodal Computing,
Max Planck Institute for Informatics, Saarbrücken, Germany
andres@mpi-inf.mpg.de

Abstract. Many computer vision problems can be cast as an optimization problem whose feasible solutions are decompositions of a graph. The minimum cost lifted multicut problem is such an optimization problem. Its objective function can penalize or reward all decompositions for which any given pair of nodes are in distinct components. While this property has many potential applications, such applications are hampered by the fact that the problem is NP-hard. We propose a fusion move algorithm for computing feasible solutions, better and more efficiently than existing algorithms. We demonstrate this and applications to image segmentation, obtaining a new state of the art for a problem in biological image analysis.

1 Introduction and Related Work

In 2011, Andres et al. [1], Bagon and Galun [2], Kim et al. [3,4] and Yarkony et al. [5] independently proposed formulating the image segmentation problem [6] as a minimum cost multicut problem [7,8] on a suitable graph. Given, for every pair of neighboring nodes, a cost or reward (negative cost) to be paid if these nodes are assigned to distinct components, the minimum cost multicut problem consists in finding a decomposition of the graph with minimal sum of costs. In 2015, Keuper et al. [9], using a construction from [10], proposed the minimum cost *lifted* multicut problem, a generalization with an identical feasible set whose objective function can assign a cost or reward to *every* pair of nodes, not just neighboring ones. These non-local interactions are represented in the graph by "lifted" edges which are subjected to slightly different constraints than the regular edges. The introduction of lifted edges is appealing for image segmentation, because non-local interactions can now be added without losing two key advantages of the multicut: (i) Every feasible solution of the optimization problem corresponds to a decomposition of the graph, i.e. to a consistent segmentation. (ii) No assumptions on the number or size of segments are made, making the method applicable in the typical and important scenario where such prior knowledge is not available. Since standard and lifted multicut are both NP-hard integer linear programming problems [7,8] – even for planar graphs [11,12]

B. Leibe et al. (Eds.): ECCV 2016, Part II, LNCS 9906, pp. 715–730, 2016.
DOI: 10.1007/978-3-319-46475-6_44

– this paper proposes a new family of efficient heuristics inspired by [13,14] and on the basis of *fusion moves* [14,15].

So far, the computer vision community has studied three classes of algorithms addressing optimization problems of this type: (i) branch-and-cut algorithms [1,16,17] that converge to an optimal integer solution but do not admit polynomial time complexity bounds and are too slow for lifted multicuts; (ii) linear programming relaxations with subsequent rounding to an integer solution [17–19] which can yield a log-factor approximation [8] in polynomial time; (iii) constrained search algorithms [9,20,21] that find approximate integer solutions directly in polynomial time. Although no theoretical guarantees are known for the latter approximations, they tend to be better than relaxation followed by rounding.

Constrained search algorithms for the lifted multicut problem were introduced in [9]. They generalize multicut algorithms of the Kernighan/Lin [22] type from [20] and greedy additive edge contraction from [21]. We show in this paper that fusion move algorithms for the multicut as proposed in [23] can be generalized as well and actually perform better in terms of approximation quality and speed.

1.1 Contribution

This work makes four contributions:

1. We generalize the fusion move algorithm [23] into a new constrained search algorithm for the minimum cost lifted multicut problem defined in [9].
2. We show that our algorithm outperforms the constrained search algorithms of [9] on the same problem instances in approximation quality and speed.
3. We introduce novel non-local potentials for the segmentation problem and incorporate them into a lifted multicut formulation of the objective.
4. We apply the proposed algorithm to the biological image segmentation benchmark [24,25], achieving the highest accuracy known at the time of writing.

2 Optimization Problem

2.1 Minimum Cost Multicut Problem

The minimum cost multicut problem is an optimization problem whose feasible solutions can be identified with the decompositions of a graph. Below, we recall only the necessary basic definitions and otherwise refer to [26,27] for details.

A *decomposition* of a graph is a partition of the node set into connected subsets. More rigorously, a decomposition of a graph $G = (V, E)$ is a partition Π of the node set V such that, for every $U \in \Pi$, the subgraph of G induced by U is connected. Every decomposition of a graph can be identified with the set of edges that straddle distinct components. Such subsets of edges are called the multicuts of the graph.

A subset $M \subseteq E$ of edges is a *multicut* of G iff there exists a decomposition Π of G such that M is the set of edges straddling distinct components. Moreover, M is a multicut of G iff no cycle in the graph intersects with M precisely once. Rigorously, for every cycle $Y \subseteq E$ of G: $|M \cap Y| \neq 1$. This characterization is intuitive: If one transitions from one component to another along the cycle, one needs to transition back before returning to the node from which one has started. It is used to state the minimum cost multicut problem:

For every graph $G = (V, E)$ and every $c : E \to \mathbb{R}$, the instance of the *minimum cost multicut problem* w.r.t. G and c is the optimization problem

$$\min_{x \in \{0,1\}^E} \quad \sum_{e \in E} c_e x_e \tag{1}$$

$$\text{subject to} \quad \forall Y \in \text{cycles}(G) \; \forall e \in Y : \; x_e \leq \sum_{e' \in Y \setminus \{e\}} x_{e'} . \tag{2}$$

2.2 Minimum Cost Lifted Multicut Problem

The minimum cost multicut problem has a limitation: A multicut makes explicit only for *neighboring* nodes whether these nodes are in distinct components of the decomposition induced by the multicut. It does not make this explicit for *non-neighboring* nodes. Thus, the cost function can introduce only for pairs of neighboring nodes a cost or reward to be paid by feasibles solutions that assign these nodes to distinct components. It cannot introduce such a cost for pairs of non-neighboring nodes. As illustrated in Fig. 1, simply considering a graph with more edges does not overcome this limitation in general.

This limitation led Andres [10] to define the minimum cost lifted multicut problem w.r.t. one graph $G = (V, E)$ whose decompositions are identified with feasible solutions, and a possibly larger graph $G' = (V, E')$ with $E \subseteq E'$ for

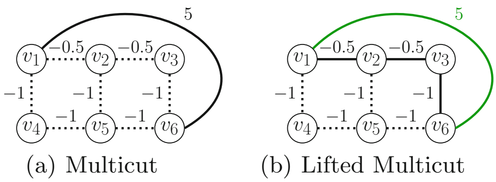


Fig. 1. Depicted above in (a) is an instance of the minimum cost multicut problem (1)–(2). The solution is the multicut consisting of those edges that are depicted as dotted lines. I.e. all edges except v_1v_6 are cut. Depicted above in (b) is an instance of the minimum cost lifted multicut problem (3)–(6) with one edge in $E' \setminus E$ depicted in green. Here as well, the solution is the lifted multicut consisting of those edges depicted as dotted lines. Note that, unlike in (a), the lifted edge with cost 5 causes the nodes v_1 and v_6 to be connected in G by a path of edges labeled 0. Thus, positive costs assigned to lifted edges are called an *attraction*.

whose every edge $vw \in E'$ it is made explicit whether the nodes v and w are in distinct components. By assigning a cost $c_{vw} \in \mathbb{R}$ to this edge, one can penalize or reward precisely those decompositions of G (!) for which the nodes v and w are in distinct components. This property is used for image segmentation in [9]. We recall the minimum cost lifted multicut problem from [10, Definition 10].

For any graphs $G = (V, E)$ and $G' = (V, E')$ with $E \subseteq E'$ and every $c : E' \to \mathbb{R}$, the instance of the *minimum cost lifted multicut problem* w.r.t. G, G' and c is the optimization problem

$$\min_{x \in \{0,1\}^{E'}} \quad \sum_{e \in E'} c_e x_e \tag{3}$$

$$\text{subject to} \quad \forall Y \in \text{cycles}(G)\ \forall e \in Y : \ x_e \leq \sum_{e' \in Y \setminus \{e\}} x_{e'} \tag{4}$$

$$\forall vw \in E' \setminus E\ \forall P \in vw\text{-paths}(G) : \ x_{vw} \leq \sum_{e \in P} x_e \tag{5}$$

$$\forall vw \in E' \setminus E\ \forall C \in vw\text{-cuts}(G) : \ 1 - x_{vw} \leq \sum_{e \in C} (1 - x_e). \tag{6}$$

The cycle constraints (4) are identical to those in (2). Additional constraints (5) and (6) ensure, for every edge $vw \in E' \setminus E$ that $x_{vw} = 0$ if (5) and only if (6) v and w are connected in G by a path of edges labeled 0, i.e., iff v and w are in the same component of G defined by the multicut $M := \{e \in E | x_e = 1\}$ of G. Or in other words, iff a lifted edge ($vw \in E' \setminus E$) is not cut, there must be a path of non-cut edges in the original graph connecting v and w.

3 Optimization Algorithm

3.1 Constrained Search Algorithms

Constrained search is a class of heuristic optimization algorithms. In the computer vision community, they are also commonly referred to as move making algorithms. Examples are α-expansion [28], $\alpha\beta$-swap [28], lazy flipping [29] and fusion [14].

Given a map $f : X \to \mathbb{R}$ and the optimization problem $\min \{f(x) \,|\, x \in X\}$, the idea of constraint search is this: Instead of optimizing f over the entire feasible set X, which might be hard, start from an initial feasible solution $x_0 \in X$, optimize f over a neighborhood $N(x_0) \subseteq X$ to obtain a new feasible solution x_1. Iff $f(x_1) < f(x_0)$, re-iterate, starting from x_1. Note that this algorithm does not require that x_1 be optimal.

Typically, the neighborhood function $N : X \to 2^X$ is chosen such that, for every $x \in X$, we have $x \in N(x)$. If N is chosen such that, for every $x \in X$, the problem $\min \{f(x') \,|\, x' \in N(x)\}$ is of polynomial time complexity, then every iteration of the algorithm is efficient. If the optimization over the neighborhood is not known to be of polynomial complexity, it can still be less complex or smaller than the original problem and can thus be tractable in practice.

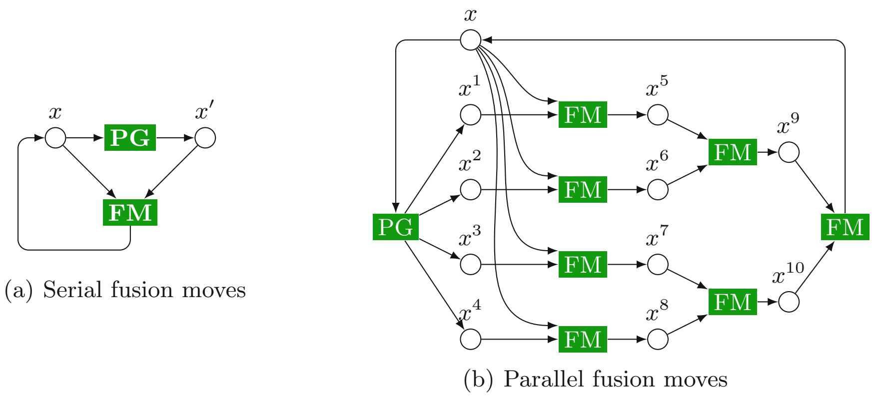


Fig. 2. In a fusion move algorithm, proposal generation (PG) and fusion moves (FM) can be combined in different ways. We implement and study serial fusion moves (a) and parallel fusion moves (b).

3.2 Fusion Move Algorithms

Fusion move algorithms [14] are a class of constrained search algorithms. They consist of two procedures. First is *proposal generation* that computes, for every feasible solution $x \in X$ given as input, another feasible solution $\textsc{pg}(x) \in X$ as output, possibly in a randomized fashion. Second is *fusion*, an optimization algorithm that computes a feasible solution of an optimization problem $\min\{f(x) \,|\, x \in N(x)\}$ for a neighborhood $N(x)$ defined w.r.t. x and $\textsc{pg}(x)$ such that $x \in N(x)$ and $\textsc{pg}(x) \in N(x)$, to obtain a feasible solution x' with $f(x') \leq f(x)$ and $f(x') \leq f(\textsc{pg}(x))$. In a fusion move algorithm, proposal generation and fusion can be combined in different ways, as depicted in Fig. 2.

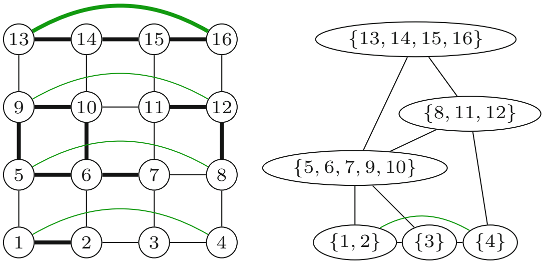
3.3 Fusion Moves for the Lifted Multicut Problem

Lempitsky introduced fusion moves for unconstrained quadratic programming in [14]. Beier et al. define a fusion move algorithm for the minimum cost multicut problem in [23]. Here, we generalize the idea of Beier et al. to the minimum cost lifted multicut problem. The fusion moves are defined in this section. Proposal generators are defined in the next section.

Given any feasible solutions x^1 and x^2 of the minimum cost lifted multicut problem (3)–(6), a constrained minimum cost lifted multicut problem in the variables $x \in \{0,1\}^{E'}$ is defined by (3)–(6) and the additional constraints

$$\forall e \in E: \quad x_e \leq x^1_e + x^2_e. \tag{7}$$

That is, all edges which are labeled 0 (join) in the feasible solution x^1 *and* the feasible solution x^2 are constrained to be labeled 0 in the problem (3)–(7). By construction, x^1 and x^2 are feasible solutions of the constrained problem (3)–(7).



(a) Graph G and E' (b) Graph $\mathcal{G}$ and $\mathcal{E}'$

Fig. 3. To perform a fusion move, we solve a minimum cost lifted multicut problem with some edge labels fixed to 0 (join). In (a) such edges are depicted by bold lines. To solve this constrained problem, we reduce it to an unconstrained minimum cost lifted problem w.r.t. a contracted graph, depicted for this example in (b).

Next, we reduce the *constrained* minimum cost lifted multicut problem (3)–(7) to an *unconstrained* minimum cost lifted multicut problem w.r.t. a smaller graph (Lemma 1). The latter problem can be solved by existing algorithms. In practice, we solve it approximatively by means of the Kernighan-Lin-type algorithm published by Keuper et al. [9]. The construction of the smaller graph is depicted in Fig. 3 and is described below.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the graph obtained from the graph G by contracting the edges $\{e \in E \,|\, x^1_e = 0 \wedge x^2_e = 0\}$[1]. Moreover, let $\mathcal{E}' \subseteq \binom{\mathcal{V}}{2}$ such that $V'W' \in \mathcal{E}'$ iff there exist $v \in V'$ and $w \in W'$ such that $vw \in E'$. Finally, let $C : \mathcal{E}' \to \mathbb{R}$ such that, for every $V'W' \in \mathcal{E}'$:

$$C_{V'W'} = \sum_{\{vw \in E' \,|\, v \in V' \wedge w \in W'\}} c_{vw} \tag{8}$$

Lemma 1. *For every feasible solution $X : \mathcal{E}' \to \{0, 1\}$ of the instance of the minimum cost lifted multicut problem w.r.t. $\mathcal{G}$, $\mathcal{G}' := (\mathcal{V}, \mathcal{E}')$ and C, the $x : E' \to \{0, 1\}$ such that*

$$\forall vw \in E' : \quad x_e = \begin{cases} X_{V'W'} & \text{if } \exists V'W' \in \mathcal{E}' : v \in V' \wedge w \in W' \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

is well-defined and a feasible solution of the constrained minimum cost lifted multicut problem (3)–(7). Moreover,

$$\sum_{vw \in E'} c_{vw} x_{vw} = \sum_{V'W' \in \mathcal{E}'} C_{V'W'} X_{V'W'}. \tag{10}$$

[1] I.e., $\mathcal{V}$ is a decomposition of G with every $V' \in \mathcal{V}$ a maximal subset $V' \subseteq V$ of nodes of G connected by edges $e \in E$ for which $x^1_e = 0$ and $x^2_e = 0$. In addition, for every $V'W' \in \binom{\mathcal{V}}{2}$, we have $V'W' \in \mathcal{E}$ iff there exist $v \in V'$ and $w \in W'$ such that $vw \in E$.

Proof. If there exist $V'W' \in \mathcal{E}'$ such that $v \in V'$ and $w \in W'$, then V' and W' are unique (because $\mathcal{V}$ is a partition of V). Thus, x is well-defined.

The feasible solution X defines a decomposition of $\mathcal{G}$ (because $\mathcal{M} := \{V'W' \in \mathcal{E} \,|\, X_{V'W'} = 1\}$ is a multicut of $\mathcal{G}$). Every decomposition of $\mathcal{G}$ induces a decomposition of G (as the node set $\mathcal{V}$ of $\mathcal{G}$ is itself a decomposition of G). The multicut $M := \{vw \in E \,|\, x_{vw} = 1\}$ of this decomposition of G is defined by the multicut $\mathcal{M}$ of $\mathcal{G}$ by (9) (by definition of $\mathcal{G}$). Thus, x satisfies (4).

Moreover, for every $vw \in E' \setminus E$, we have $x_{vw} = 0$ iff v is connected to w by a path P in G with $x_P = 0$ (by (9) and definition of $\mathcal{G}$ and $\mathcal{E}'$). Thus, x satisfies (5) and (6). Finally, (10) holds by (8) and (9). □

3.4 Proposal Generation for the Lifted Multicut Problem

As pointed out in [30], a proposal generator is designed with four objectives in mind. Firstly, proposed feasible solutions should be *diverse*. Otherwise, the fusion move algorithms can get trapped in local minima. Secondly, some proposed feasible solutions should be *good*. Otherwise, the fusion move algorithms cannot get close to the optimum. In the context of the minimum cost lifted multicut problem, a feasible solution is good if the recall of edges that are cut in an optimal solution is close to 1. Thirdly, the proposed feasible solutions should be *sparse*. In the context of the minimum cost lifted multicut problem, a feasible solutions is sparse if the precision of edges that are cut in an optimal solution is close to 1. Fourthly, the proposed feasible solutions should be *cheap*, i.e., proposals should be computable efficiently and in parallel. We study three proposal generators that emphasize different design objectives.

Randomly Perturbed Proposals. In order to obtain a proposal of high quality efficiently, we apply greedy additive edge contraction (GAEC) [9]. The key idea of this algorithm is to greedily contract edges with maximum cost until this maximum cost is equal to or smaller than zero. In order to get diverse solutions, we follow [23] and add normally distributed noise of zero mean to edge costs. In order to control the sparsity of the proposal, we replace the stopping criterion of GAEC and continue until a maximum allowed number of components is reached.

Subgraph Proposals. In order to obtain an objective-aware proposal for a large problem instance, we solve the minimum cost lifted multicut problem for a small subgraph. Technically, the procedure works as follows: We choose a center node $v \in V$ and the subgraph induced by the set U of all nodes within a fixed path-length distance from v. For $E_0 := \{vw \in E \,|\, v \notin U \wedge w \notin U\}$ and $E_1 := \{vw \in E \,|\, v \in U \wedge w \notin U\}$, we solve the instance of the minimum cost lifted multicut problem w.r.t. the graph G and the cost function c, with the additional constraints

$$\forall e \in E_0 : \quad x_e = 0 \tag{11}$$

$$\forall e \in E_1 : \quad x_e = 1. \tag{12}$$

Watershed Proposals. In order to obtain diverse proposals cheaply, we follow [23] in using the weighted watershed algorithm [31,32] with random seeds. From the set $\{vw \in E' \setminus E \,|\, c_{vw} < 0\}$ of lifted edges with negative cost, we draw a fixed number without replacement and assign different seeds to v and w. Thus, a random subset of lifted edges with negative cost is cut.

4 Experiments

We now describe experiments in which we compare the fusion move algorithm (Algorithm 1) for the minimum cost lifted multicut problem with the Kernighan/Lin-type algorithm (KLj) and Greedy Additive Edge Contraction (GAEC) of [9] for the same problem.

In the tables below, FM-R, FM-SG and FM-WS stand for the fusion move algorithm with the randomized, subgraph and watershed proposal generators, respectively. Individual fusion problems, i.e., those problems denoted by boxes labeled "FM" in Fig. 2, are solved by KLj initialized with the output of GAEC.

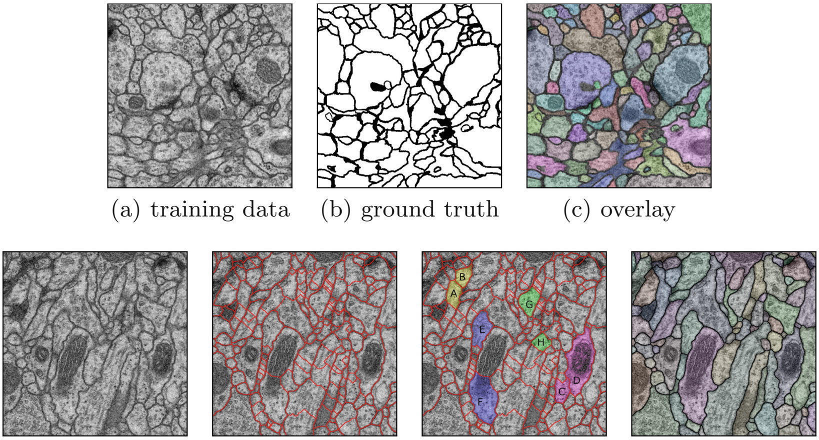
In each experiment, the outer loop of fusion is terminated when no improvement is achieved for 5 consecutive iterations. Each experiment is conducted with 1, 2, 4 and 8 threads, respectively, to examine concurrency. All experiments are conducted on an Intel Core i7-4700MQ CPU operating at 2.40 GHz $\times$ 8, and equipped with 32 GB of RAM.

4.1 ISBI 2012 Challenge

The ISBI 2012 Challenge [24,25] offers a set of segmentation tasks where images of the Drosophila larva ventral nerve cord acquired by a serial section

Data: $\boldsymbol{G}$: Graph $G = (V, E)$; $\boldsymbol{G'}$: Graph $G' = (V, E')$
$\boldsymbol{C}$: edge weights; $\boldsymbol{x_{start}}$: starting point solution; $\boldsymbol{GEN}$: proposal generator
Result: $\boldsymbol{y}$: improved solution
$x_{best} \leftarrow x_{start}$;
for < *nIterations* > **do**
  $P \leftarrow \emptyset$;
  for *#Proposals in parallel* **do**
    $x^p \leftarrow \mathrm{GEN}(x_{best})$ ▷ generate proposal;
    $x^p \leftarrow \arg\min_{x \in \{0,1\}} \sum_{e \in E'} c_e x_e \quad \text{s.t. (3)–(7)}, \forall e \in E: \quad x_e \le x_e^p + x_e^{best}$.;
    $P \leftarrow P \cup \{x^p\}$;
  end
  while $|P| > 1$ **do**
    ▷ hierarchically fuse proposals ;
    $\hat{P} \leftarrow \emptyset$;
    for *each i in* $|P|/2$ *in parallel* **do**
      $x^1 \leftarrow P_{2i}$;
      $x^2 \leftarrow P_{2i+1}$;
      $x^p \leftarrow \arg\min_{x \in \{0,1\}} \sum_{e \in E'} c_e x_e \quad \text{s.t. (3)–(7)}, \forall e \in E: \quad x_e \le x_e^1 + x_e^2$.;
      $\hat{P} \leftarrow \hat{P} \cup \{x^p\}$;
    end
    $P \leftarrow \hat{P}$;
  end
  $x_{best} \leftarrow P_1$ ▷ update current best
end
return x_{best};

Algorithm 1. Lifted MC - Parallel Fusion Moves (LMC-PFM)



(a) training data (b) ground truth (c) overlay

(d) test data (e) superpixel (f) local and lifted edges in the RAG (g) result

Fig. 4. The ISBI 2012 Challenge [24,25] offers a set of segmentation tasks where neurons are to be delineated correctly in two-dimensional electron microscopy images, cf. (a)–(c). We start from the region adjacency graph of a superpixel segmentation (e) and train two classifiers to estimate the probability of adjacent and, respectively, non-adjacent superpixel pairs to belong to the same neuron. I.e., for edges like A-B and C-D in (f) or lifted edges E-F and G-H in (f). Solving, by fusion moves, a minimum cost multicut problem with costs defined in (13), our results on independent test images (with undisclosed ground truth) achieve the highest accuracy known at the time of writing. See (g) and Table 1.

transmission electron microscope are to be decomposed into distinct neurons, as depicted in Fig. 4c. The data set contains of 30 training images and 30 test images. Human annotations (Fig. 4b) are provided for each training image.

We propose a processing pipeline. Describing this pipeline in every technical detail is beyond the scope of this work. For the sake of reproducibility, the source code is available[2]. Overall, the pipeline consists of the following steps:

1. Start from the region adjacency graph (RAG) of an over-segmentation generated by seeded region growing [33], as shown in Fig. 4e.
2. Add lifted edges F for all pairs of superpixels within a path-length distance of $r_{nl} = 4$. The difference between lifted and non-lifted edges can be seen in Fig. 4f.
3. Train two random forest classifiers: A first classifier RF_l learns to predict if a pair of adjacent superpixels should be in the same neuron or not. A second classifier RF_{nl} predicts the same for non-adjacent pairs of superpixels.
4. Solve an instance of the minimum cost lifted multicut problem (3)–(6) with superpixels as nodes, non-lifted and lifted edges and costs defined w.r.t. the

[2] https://github.com/DerThorsten/lifted_fusion_moves_eccv_2016.

probabilities estimated by RF_l and RF_{nl} as

$$c_{vw} := \log \frac{p(x_{vw} = 0)}{p(x_{vw} = 1)}. \tag{13}$$

To train RF_l we use features on local image statistics as described in [34,35]. To train RF_{nl}, we compute the following features for of lifted edges:

1. Features based on hierarchical clustering inspired by [36,37]: We apply UCM to generate the complete dendrogram and use the thus defined ultrametric distance between pairs of nodes (height in the dendrogram at the moment when the nodes are merged) as a feature for the corresponding lifted edge, if it exists.
2. Features inspired by maximum intervening contours [38–40]: We compute simple statistics of local image features (e.g. average gradient) along multiple straight lines between two superpixels.
3. Shortest path based features: Using various local features (raw intensities, gradients etc.), we compute multiple shortest paths between non-adjacent superpixels and measure statistics along these paths.
4. Candidate segmentation features: We compute multiple candidate segmentations using the minimum multicut objective (with varying parameter and without lifted edges), and each edge is assigned the proportion of the segmentation where it got cut.

For all features above we use the raw data itself as input, but also a pixel wise probability map learned with a CNN [41].

A quantitative evaluation is shown in Table 1. It can be seen from this table that segmentations of the images defined by feasibles solutions of the minimum cost lifted multicut problem define a new state of the art on this highly competitive segmentation challenge. FM-R, FM-SG and KLj yield the same objective.

Table 1. Feasible solutions of the minimum cost lifted multicut problem define a new state of the art on the ISBI 2012 Challenge [24,25]. The performance measures VRand and VInfo are defined in [25]. A value of 1 indicates a perfect segmentation; values close to zero indicate poor segmentations. Using 8 threads, the proposed methods (FM-R, FM-SG) outperform KLj by a factor of 4. Leader board: http://brainiac2.mit.edu/isbi_challenge/leaders-board-new

Algorithm	Objective	Time to convergence [s] (1/2/4/8 threads)	VRand	VInfo
			(Higher is better)	
FM-SG	−13560.18	0.62/0.37/0.28/0.21	0.9804	0.9884
FM-R	−13560.18	0.77/0.42 / 0.32/0.28	0.9804	0.9884
KLj	−13560.18	0.89	0.9803	0.9884
Leader Board 2	-	-	0.9796	0.9870
Leader Board 3	-	-	0.9768	0.9886
Humans	-	-	0.9978	0.9990

Even with only a single thread, FM-R and FM-SG are slightly faster than KLj. With 8 threads, the proposed methods outperform KLj by a factor of 4.

4.2 Image Decomposition

Keuper et al. [9] pose the image decomposition problem [6] as a minimum cost lifted multicut problem. Instances of this problem are defined w.r.t. pixel grid graphs and lifted edges connecting each pixel to the (about 300) pixels within a path-length distance of 10. Costs of non-lifted edges are derived from structured edge detection according to [42]. Costs of lifted edges are defined by probabilistic geodesic lifting [9].

These large instances of the minimum cost multicut problem pose a challenge to optimization algorithms and are thus suitable for benchmarking. Here, we compare the fusion move algorithm with watershed proposal generator (FM-WS) with GAEC and KLj initialized with the output of GAEC.

Results are shown in Table 2. It can be seen from these results that FM-WS outperforms the current state of the art (KLj) in terms of runtime and objective value. Moreover, FM-WS is about twice as fast with one thread and about six times as fast with 8 threads. The gap between FM-WS and KLj is comparatively larger than that between of KLj and GAEC. Therefore, we consider FM-WS a significant improvement over the state of the art.

Table 2. The proposed algorithm FM-WS outperforms KLj and GAEC on the large and hard instances of the minimum cost lifted multicut problem of [9].

Algorithm	Objective	Time to convergence [s] (1,2,4,8 threads)
FM-WS	−62748200	61/32/25/22
GAEC	−62744700	10/n.a
KLj	−62745500	121/n.a

4.3 Averaging Multiple Segmentations

Fusing multiple segmentations into a single one is not only important as an image analysis sub-task, but can also be used to combine multiple *manually* derived ground truth solutions into a "master" ground truth image. Multiple user-provided solutions are, for example, available for the BSDS-500 data set [6].

Recently, [43] proposed to solve this problem with an EM-algorithm based on the multicut objective. Their algorithm is defined on a complete graph derived from the region adjacency graph of an initial superpixel segmentation. In contrast to our approach, they use the plain multicut objective where all edges of the complete graph are considered local, and there are no lifted edges. Before constructioning the complete graph, every proposed segmentation x^l from the given set L is projected on the superpixel RAG, and all edges which are not cut in any proposal are contracted, resulting in a dramatic reduction of the graph's size. The edge costs of the remaining edges measure how often this edge is cut in L. Furthermore, a weight p_l measuring the estimated reliability of each segmentation relative to the others is assigned to each member of L. The multicut objective is then optimized with p_l kept fixed, and the p_l are updated according to the proportion of edges in x^l that agree with the current master segmentation. This is repeated in an EM manner until convergence.

We modify this approach as follows: We optimize directly on the *pixel-level*, i.e. on a 4-connected grid graph instead of a superpixel RAG, to eleminate superpixel computation as an additional source of error. Moreover, we replace the multicut objective with a *lifted multicut* objective containing only a sparse set of lifted edges up to a graph distance of 5. We do not contract any edges in pre-processing. Edge costs are defined as in [43] by

$$c_{vw} := \log \sum_{l \in |L|} (1 - x^l_{vw}) p_l - \log \sum_{l \in |L|} x^l_{vw} p_l \tag{14}$$

As in [43], we use an EM-type algorithm to update p_l according to the number of edges in x^l that agree with the current master segmentation $\hat{x}$:

$$p_l = \frac{1}{|EF|} \sum_{x_{vw} \in EV} 1 - |x^l_{vw} - \hat{x}_{vw}| \tag{15}$$

In every iteration of EM, we solve an instance of the minimum cost lifted multicut problem using FM-SG and, for comparison, KLj. Both are initialized with the output of GAEC. We only use FM-SG results to update the p_l since they were always better than the KL results. In addition to the proposals generated by the subgraph method, all x^l are included into the proposal set, leading to a significant speed-up.

Results are shown in Table 3 and Fig. 5. It can be seen from Table 3 that FM-SG outperforms KLj in terms of objective value and run-time. Even with a single thread, FM-SG is twice as fast as KL. Using 8 threads, the FM-SG is six times as fast.

Table 3. To average multiple segmentations, we solve instances of a minimum cost lifted multicut problem as part of the EM algorithm proposed in [43]. FM-SG is an efficient algorithm to solve these instances.

Algorithm	Objective	Time to convergence [s] (1/2/4/8 Threads)
FM-SG	**−2.29e+07**	14.8/8.83/6.33/5.21
GAEC	−1.53e+07	13.8
GAEC + KLj	−2.27e+07	29.3

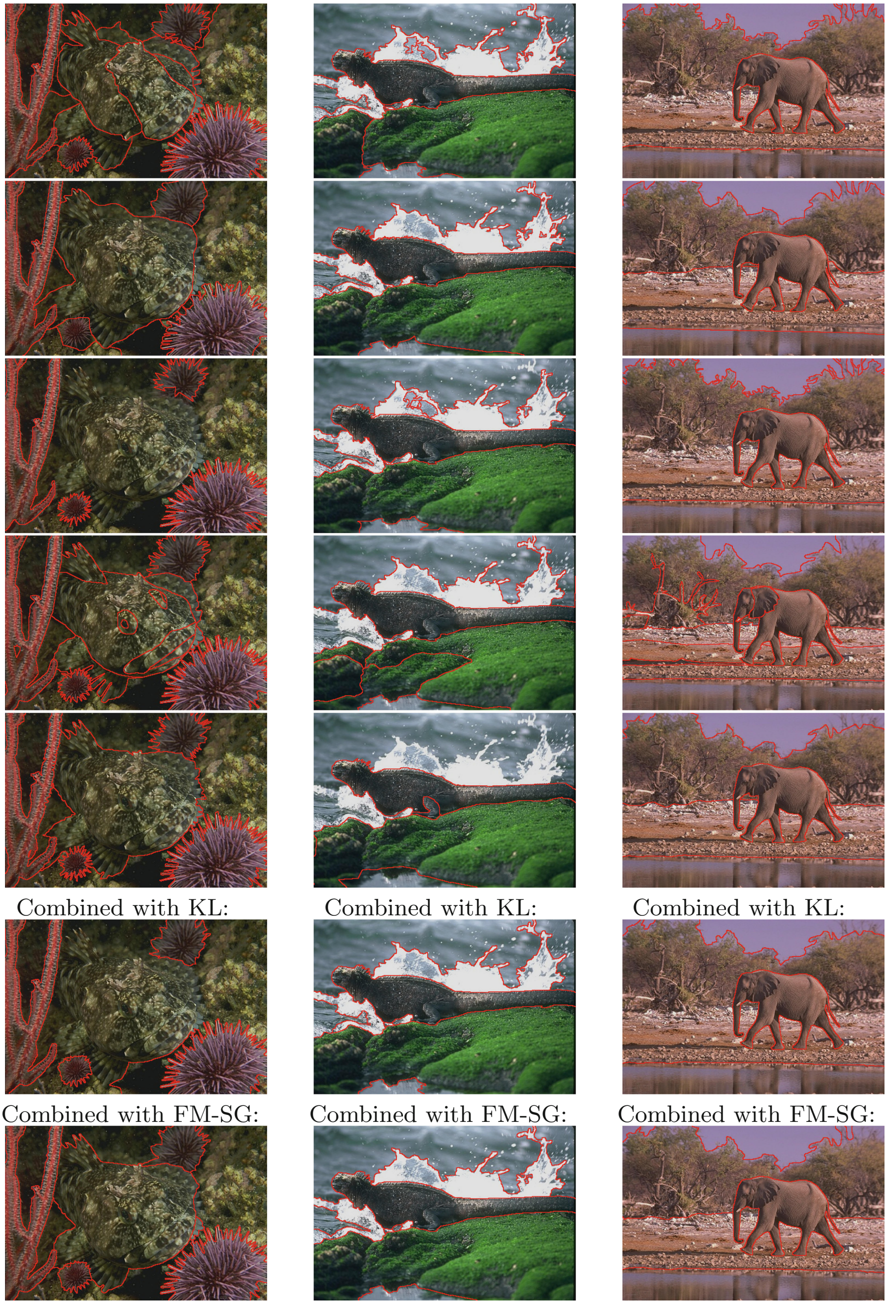


Fig. 5. To average multiple segmentations, we solve instances of a minimum cost lifted multicut problem as in (14)–(15). Above, Rows 1–5 show different man-made segmentations of images from the BSDS-500 benchmark [6]. Row 6 shows the combination of these segmentations by the solution using KL, row 7 shows the result with the proposed algorithm (FM-SG).

5 Conclusion

We have defined a fast, scalable and easy to implement fusion move algorithm for the minimum cost lifted multicut problem. Experiments with diverse instances of the problem have shown that this algorithm typically outperforms existing methods in terms of objective value and run-time. We conjecture that efficient algorithms such as the one proposed in this paper facilitate a variety of applications of the minimum cost lifted multicut problem in computer vision of which the averaging of multiple segmentations is just one example. Improved parallelization schemes of the proposed algorithm are subject of future work.

Acknowledgments. Partial financial support by DFG-SFB 1129, DFG-SFB 1134 and DFG-HA 4364/8-1 is gratefully acknowledged.

References

1. Andres, B., Kappes, J.H., Beier, T., Köthe, U., Hamprecht, F.A.: Probabilistic image segmentation with closedness constraints. In: ICCV (2011)
2. Bagon, S., Galun, M.: Large scale correlation clustering optimization. CoRR abs/1112.2903 (2011). http://arxiv.org/abs/1112.2903
3. Kim, T., Nowozin, S., Kohli, P., Yoo, C.D.: Variable grouping for energy minimization. In: CVPR (2011)
4. Kim, S., Yoo, C., Nowozin, S., Kohli, P.: Image segmentation using higher-order correlation clustering. TPAMI **36**, 1761–1774 (2014)
5. Yarkony, J., Ihler, A., Fowlkes, C.C.: Fast planar correlation clustering for image segmentation. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VI. LNCS, vol. 7577, pp. 568–581. Springer, Heidelberg (2012)
6. Arbeláez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **33**, 898–916 (2011)
7. Bansal, N., Blum, A., Chawla, S.: Correlation clustering. Mach. Learn. **56**(1–3), 89–113 (2004)
8. Demaine, E.D., Emanuel, D., Fiat, A., Immorlica, N.: Correlation clustering in general weighted graphs. Theoret. Comput. Sci. **361**(2–3), 172–187 (2006)
9. Keuper, M., Levinkov, E., Bonneel, N., Lavoué, G., Brox, T., Andres, B.: Efficient decomposition of image and mesh graphs by lifted multicuts. In: ICCV (2015)
10. Andres, B.: Lifting of multicuts. CoRR abs/1503.03791 (2015). http://arxiv.org/abs/1503.03791
11. Voice, T., Polukarov, M., Jennings, N.R.: Coalition structure generation over graphs. J. Artif. Intell. Res. **45**, 165–196 (2012)
12. Bachrach, Y., Kohli, P., Kolmogorov, V., Zadimoghaddam, M.: Optimal coalition structure generation in cooperative graph games. In: AAAI (2013)
13. Cook, W., Seymour, P.: Tour merging via branch-decomposition. INFORMS J. Comput. **15**(3), 233–248 (2003)
14. Lempitsky, V.S., Rother, C., Roth, S., Blake, A.: Fusion moves for Markov random field optimization. IEEE Trans. Pattern Anal. Mach. Intell. **32**(8), 1392–1405 (2010)

15. Kappes, J.H., Beier, T., Schnörr, C.: MAP-inference on large scale higher-order discrete graphical models by fusion moves. In: International Workshop on Graphical Models in Computer Vision (2014)
16. 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: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part III. LNCS, vol. 7574, pp. 778–791. Springer, Heidelberg (2012)
17. Kappes, J.H., Speth, M., Reinelt, G., Schnörr, C.: Higher-order segmentation via multicuts. Comput. Vis. Image Underst. **143**, 104–119 (2016). Inference and Learning of Graphical Models Theory and Applications in Computer Vision and Image Analysis
18. Kappes, J.H., Speth, M., Andres, B., Reinelt, G., Schn, C.: Globally optimal image partitioning by multicuts. In: Boykov, Y., Kahl, F., Lempitsky, V., Schmidt, F.R. (eds.) EMMCVPR 2011. LNCS, vol. 6819, pp. 31–44. Springer, Heidelberg (2011)
19. Yarkony, J., Ihler, A., Fowlkes, C.C.: Fast planar correlation clustering for image segmentation. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VI. LNCS, vol. 7577, pp. 568–581. Springer, Heidelberg (2012)
20. Andres, B., Beier, T., Kappes, J.H.: OpenGM: a C++ library for discrete graphical models. arXiv e-prints (2012)
21. Beier, T., Kroeger, T., Kappes, J.H., Koethe, U., Hamprecht, F.: Cut, glue & cut: a fast, approximate solver for multicut partitioning. In: IEEE Conference on Computer Vision and Pattern Recognition 2014 (2014)
22. Kernighan, B.W., Lin, S.: An efficient heuristic procedure for partitioning graphs. Bell Syst. Tech. J. **49**(2), 291–308 (1970)
23. Beier, T., Hamprecht, F.A., Kappes, J.H.: Fusion moves for correlation clustering. In: CVPR Proceedings (2015, in press)
24. Cardona, A., Saalfeld, S., Preibisch, S., Schmid, B., Cheng, A., Pulokas, J., Tomancak, P., Hartenstein, V.: An integrated micro- and macroarchitectural analysis of the Drosophila brain by computer-assisted serial section electron microscopy. PLoS Biol. **8**(10), 1–17 (2010)
25. Arganda-Carreras, I., Turaga, S.C., Berger, D.R., Ciresan, D., Giusti, A., Gambardella, L.M., Schmidhuber, J., Laptev, D., Dwivedi, S., Buhmann, J.M., Liu, T., Seyedhosseini, M., Tasdizen, T., Kamentsky, L., Burget, R., Uher, V., Tan, X., Sun, C., Pham, T., Bas, E., Uzunbas, M.G., Cardona, A., Schindelin, J., Seung, H.S.: Crowdsourcing the creation of image segmentation algorithms for connectomics. Front. Neuroanat. **9**(142), 1–13 (2015)
26. Grötschel, M., Wakabayashi, Y.: A cutting plane algorithm for a clustering problem. Math. Program. **45**(1), 59–96 (1989)
27. Chopra, S., Rao, M.: The partition problem. Math. Program. **59**(1–3), 87–115 (1993)
28. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? IEEE Trans. Pattern Anal. Mach. Intell. **26**(2), 147–159 (2004)
29. Andres, B., Kappes, J.H., Beier, T., Köthe, U., Hamprecht, F.A.: The lazy flipper: efficient depth-limited exhaustive search in discrete graphical models. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VII. LNCS, vol. 7578, pp. 154–166. Springer, Heidelberg (2012)
30. Lempitsky, V., Rother, C., Roth, S., Blake, A.: Fusion moves for Markov random field optimization. IEEE Trans. Pattern Anal. Mach. Intell. **32**(8), 1392–1405 (2010)

31. Meyer, F.: Watersheds on edge or node weighted graphs “par l’exemple”. CoRR abs/1303.1829 (2013). http://arxiv.org/abs/1303.1829
32. Meyer, F.: Stochastic watershed hierarchies. In: ICAPR 2015; 8th International Conference on Advances in Pattern Recognition. Indian Statistical Institute, Kolkata, January 2015
33. Beucher, S., Meyer, F.: The morphological approach to segmentation: the watershed transformation. Opt. Eng. **34**, 433–433 (1992)
34. Andres, B., Köthe, U., Kroeger, T., Helmstaedter, M., Briggman, K.L., Denk, W., Hamprecht, F.A.: 3D segmentation of SBFSEM images of neuropil by a graphical model over supervoxel boundaries. Med. Image Anal. **16**(4), 796–805 (2012)
35. 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: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part III. LNCS, vol. 7574, pp. 778–791. Springer, Heidelberg (2012)
36. Arbelaez, P.: Boundary extraction in natural images using ultrametric contour maps. In: Proceedings of 2006 Conference on Computer Vision and Pattern Recognition Workshop, CVPRW 2006, p. 182. IEEE Computer Society, Washington, DC (2006)
37. Yang, X., Prasad, L., Latecki, L.J.: Affinity learning with diffusion on tensor product graph. IEEE Trans. Pattern Anal. Mach. Intell. **35**(1), 28–38 (2013)
38. Leung, T., Malik, J.: Contour continuity in region based image segmentation. In: Burkhardt, H.-J., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1406, p. 544. Springer, Heidelberg (1998)
39. Fowlkes, C., Malik, J.: How much does globalization help segmentation? Technical report, Division of Computer Science, University of California, Berkeley, July 2004
40. Maire, M., Arbelaez, P., Fowlkes, C.C., Malik, J.: Using contours to detect and localize junctions in natural images. In: CVPR (2008)
41. Lin, M., Chen, Q., Yan, S.: Network in network. CoRR abs/1312.4400 (2013). http://arxiv.org/abs/1312.4400
42. Dollár, P., Zitnick, C.L.: Fast edge detection using structured forests. PAMI **37**, 1558–1570 (2015)
43. Alush, A., Goldberger, J.: Ensemble segmentation using efficient integer linear programming. Pattern Anal. Mach. Intell. **34**(10), 1966–1977 (2012)