

# Efficient Decomposition of Image and Mesh Graphs by Lifted Multicuts

M. Keuper<sup>1</sup>, E. Levinkov<sup>2</sup>, N. Bonneel<sup>3</sup>, G. Lavoué<sup>3</sup>, T. Brox<sup>1</sup> and B. Andres<sup>2,\*</sup>

<sup>1</sup>Department of Computer Science, University of Freiburg

<sup>2</sup>Combinatorial Image Analysis, MPI for Informatics, Saarbrücken

<sup>3</sup>Laboratoire d’Informatique en Image et Systèmes d’Information, CNRS Lyon

## Abstract

*Formulations of the Image Decomposition Problem [8] as a Multicut Problem (MP) w.r.t. a superpixel graph have received considerable attention. In contrast, instances of the MP w.r.t. a pixel grid graph have received little attention, firstly, because the MP is NP-hard and instances w.r.t. a pixel grid graph are hard to solve in practice, and, secondly, due to the lack of long-range terms in the objective function of the MP. We propose a generalization of the MP with long-range terms (LMP). We design and implement two efficient algorithms (primal feasible heuristics) for the MP and LMP which allow us to study instances of both problems w.r.t. the pixel grid graphs of the images in the BSDS-500 benchmark [8]. The decompositions we obtain do not differ significantly from the state of the art, suggesting that the LMP is a competitive formulation of the Image Decomposition Problem. To demonstrate the generality of the LMP, we apply it also to the Mesh Decomposition Problem posed by the Princeton benchmark [16], obtaining state-of-the-art decompositions.*

## 1. Introduction

Formulations of the Image Decomposition Problem [8] as a Minimum Cost Multicut Problem (MP) [17, 19] have received considerable attention [2, 5, 6, 7, 10, 12, 13, 26, 27, 28, 31, 32, 35, 40, 41]. Advantages of this formulation are in order: Firstly, the feasible solutions of the MP relate one-to-one to the decompositions of a graph. In particular, the number of components is not fixed in advance but is determined by the solution. Secondly, the MP, unlike balanced cut problems [37], does not favor one decomposition over another by definition. Thirdly, multicut algorithms are easy to use; they take as input a graph, *e.g.* the pixel grid graph of an image, and, for every edge, a real-valued cost (reward) of the incident nodes being in distinct components, *e.g.*  $\log \frac{1-p_e}{p_e} + \log \frac{1-p^*}{p^*}$ , for an estimated probability  $p_e$  of boundary [8] at the edge  $e$ , and a prior probability  $p^* \in (0, 1)$

of cuts. The output is a 01-labeling of the edges that well-defines a decomposition of the graph by 0 indicating “join” and 1 indicating “cut”.

One disadvantage is the NP-hardness of the MP [11, 18]. Despite significant progress in the design of efficient heuristics [10, 12, 13, 29], instances of the MP for image segmentation have so far only been solved w.r.t. superpixel adjacency graphs and not w.r.t. pixel grid graphs, with the sole and notable exception of [10]. A second disadvantage results from the fact that a multicut makes explicit only for edges whether the incident nodes are in distinct components. It does not make explicit for pairs of nodes that are not neighbors whether these are in distinct components. Hence, the linear objective function of the MP w.r.t. a pixel grid graph cannot assign a cost specifically to all decompositions for which a pair of pixels that are not neighbors are in distinct components. This limitation, noted *e.g.* in [7], hampers applications as it is often hard to estimate, for an image and a pair of neighboring pixels, whether the image is to be cut precisely between these pixels (only these estimates are used in the MP), and as it is sometimes easy to estimate for pixels at larger distance whether these are in distinct components (these estimates are not used in the MP).

An optimization problem whose feasible solutions relate one-to-one to the decompositions of a graph and whose objective function can assign, for any pair of nodes, a cost to all decompositions for which these nodes are in distinct components, although desirable, has not been proposed before.

**Contribution.** We propose the Minimum Cost Lifted Multicut Problem (LMP), a generalization of the MP whose feasible solutions relate one-to-one to the decompositions of a graph and whose objective function can assign, for any pair of nodes, a real-valued cost (reward) to all decompositions for which these nodes are in distinct components. We design and implement two efficient algorithms for both the MP and the LMP and evaluate both problem formulations in conjunction with both algorithms for the Image Decomposition Problem in terms of the BSDS-500 benchmark [8] and for the Mesh Decomposition Problem in terms of the Princeton Mesh Segmentation benchmark [16].

\*Correspondence: [andres@mpi-inf.mpg.de](mailto:andres@mpi-inf.mpg.de)

## 2. Related Work

The MP is known as Correlation Clustering in machine learning and theoretical computer science [11, 18]. For complete graphs, which are of special interest in machine learning, the well-known MP and the proposed LMP coincide.

A generalization of the MP by a higher-order objective function, called the Higher-Order Multicut Problem (HMP), was proposed in [31] and is studied in detail in [27, 32]. In principle, the HMP subsumes all optimization problems whose feasible solutions coincide with the multicuts of a graph, including the LMP we propose. In fact, the HMP is strictly more general than the LMP; its objective function can assign an objective value to all decompositions for which any set of edges is cut, unlike the objective function of the LMP which is limited to single edges. However, the instances of the HMP that are equivalent to the instances of the LMP we propose have an objective function whose order is equal to the number of edges in the graph and are hence impractical. Thus, the HMP and LMP are complementary in practice.

Efficient algorithms (primal feasible heuristics) for the MP are proposed and analyzed in [10, 13, 12, 29]. The algorithms we design and implement are compared here to the state of the art [13]. Our implementation of (an extension of) the Kernighan-Lin Algorithm (KL) [29] is compared here, in addition, to the implementation of KL in [4, 25].

Toward image decomposition [8], the state of the art in boundary detection is [15, 21], followed closely by [20, 23]. Our experiments are based on [20] which is publicly available and outperformed marginally by [15, 21]. The state of the art in image decomposition is [9], followed closely by [8, 23]. Our results are compared quantitatively to [9].

Toward mesh decomposition [38], the state of the art is [24, 14, 39], followed closely by [42, 33]. Our experiments are based on [24, 42, 33]. In prior work, methods based on learning mostly rely on a unary term which requires components to be labeled semantically [24, 39]. One method based on edge probabilities was introduced previously [14]. It applies a complex post-process (contour thinning and completion, snake movement) to obtain a decomposition. We show the first mesh decompositions based on multicuts.

## 3. Problem Formulation

### 3.1. Minimum Cost Lifted Multicut Problem

We now define an optimization problem, the Minimum Cost Lifted Multicut Problem, whose feasible solutions relate one-to-one to the decompositions of a graph and whose objective function can assign, for any pair of nodes, a cost to all decompositions for which these nodes are in distinct components. Here, a component of a graph is any non-empty subgraph that is node-induced and connected. A decomposition of a graph is any partition  $\Pi$  of the node set such that, for every  $V' \in \Pi$ , the subgraph induced by  $V'$  is connected

(and hence a component of the graph). An instance of the problem is defined w.r.t.:

- A simple, undirected graph  $G = (V, E)$ , e.g., the pixel grid graph of an image or the triangle adjacency graph of a mesh.
- Additional edges  $F \subseteq \binom{V}{2} \setminus E$  connecting nodes that are not neighbors in  $G$ . In practice, we choose  $F$  so as to connect any two nodes  $v, w \in V$  whose distance  $d_{vw}$  in the graph holds  $1 < d_{vw} \leq d^*$  for a maximum distance  $d^* \in \mathbb{R}_0^+$ , fixed for the experiments in Sec. 5.
- For every edge  $vw \in E \cup F$ , a cost  $c_{vw} \in \mathbb{R}$  assigned to all feasible solutions for which  $v$  and  $w$  are in distinct components. The estimation of  $c_{vw}$  from image and mesh data is discussed in Sections 3.3 and 5.

With respect to the above, we define a feasible set  $Y_{EF} \subseteq \{0, 1\}^{E \cup F}$  whose elements  $y \in Y_{EF}$  are 01-labelings of all edges  $E \cup F$ . The feasible set is defined such that two conditions hold: Firstly, the feasible solutions  $y \in Y_{EF}$  relate one-to-one to the decompositions of the graph  $G$ . Secondly, for every edge  $vw \in E \cup F$ ,  $y_{vw} = 1$  if and only if  $v$  and  $w$  are in distinct components of  $G$ . This is expressed rigorously by two classes of constraints: The linear inequalities (2) below constrain  $y$  such that  $\{e \in E \mid y_e = 1\}$  is a multicut of the graph  $G$  [17]. For any decomposition of a graph, the multicut related to the decomposition is the subset of those edges that straddle distinct components. In addition, the linear inequalities (3) and (4) constrain  $y$  such that, for any  $vw \in F$ ,  $y_{vw} = 0$  if and only if there exists a path in  $G$  from  $v$  to  $w$ , along which all edges are labeled 0.

**Definition 1** For any simple, undirected graph  $G = (V, E)$ , any  $F \subseteq \binom{V}{2} \setminus E$  and any  $c : E \cup F \rightarrow \mathbb{R}$ , the 01 linear program written below is called an instance of the *Minimum Cost Lifted Multicut Problem (LMP)* w.r.t.  $G, F$  and  $c$ .

$$\min_{y \in Y_{EF}} \sum_{e \in E \cup F} c_e y_e \quad (1)$$

with  $Y_{EF} \subseteq \{0, 1\}^{E \cup F}$  the set of all  $y \in \{0, 1\}^{E \cup F}$  with

$$\forall C \in \text{cycles}(G) \forall e \in C : y_e \leq \sum_{e' \in C \setminus \{e\}} y_{e'} \quad (2)$$

$$\forall vw \in F \forall P \in vw\text{-paths}(G) : y_{vw} \leq \sum_{e \in P} y_e \quad (3)$$

$$\forall vw \in F \forall C \in vw\text{-cuts}(G) : 1 - y_{vw} \leq \sum_{e \in C} (1 - y_e) \quad (4)$$

### 3.2. Properties

We now discuss properties of the LMP (Def. 1):

For  $F = \emptyset$ , the LMP specializes to the MP [17, 19]. Its feasible set  $Y_{E\emptyset}$  consists of the characteristic functions of

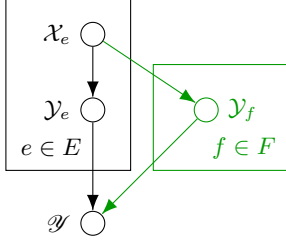


Figure 1. Depicted in black is a Bayesian Network defining a set of probability measures on multicuts [5]. Depicted in green is our extension defining a set of probability measures on lifted multicuts.

all multicuts of  $G$  (which relate one-to-one to the decompositions of  $G$ ). Its linear objective function can be chosen so as to assign, for any edge  $vw \in E$ , a cost  $c_{vw} \in \mathbb{R}$  to all decompositions of  $G$  for which the nodes  $v$  and  $w$  are in distinct components. It cannot be chosen so as to assign, for distinct nodes  $v$  and  $w$  that are not neighbors in  $G$ , a cost precisely to all decompositions of  $G$  for which  $v$  and  $w$  are in distinct components.

For  $F \neq \emptyset$ , the LMP is not a MP. Its feasible solutions still relate one-to-one to the decompositions of the graph  $G$  (because  $\varphi : Y_{EF} \rightarrow Y_{E\emptyset} : y \mapsto y_E$  is a bijection). Its objective function can be chosen so as to assign, for any  $vw \in E \cup F$ , a cost to all decompositions for which the nodes  $v$  and  $w$  are in distinct components. Thus, the LMP generalizes the MP. The feasible solutions  $y \in Y_{EF}$  are called *lifted multicuts* from  $(V, E)$  to  $(V, E \cup F)$  and are studied in [3].

For some instances of the LMP, notably if  $c_F < 0$  [7], its solutions can be identified with the solutions of the instance of the MP w.r.t. the larger graph  $G' := (V, E \cup F)$  and  $c$ . For the general LMP, this is not true. The feasible solutions of the MP with respect to  $G'$  and  $c$  do not relate one-to-one to the decompositions of  $G$ , unlike the feasible solutions of the LMP which are the characteristic functions of *some* multicuts of  $G'$ , namely those that are *lifted* from  $G$  [3].

A cutting plane algorithm for the LMP, based on the canonical LP-relaxation of the ILP in Def. 1, is impractical for the instances we consider in Sec. 5: Although the inequalities (2)–(4) can be separated efficiently, the number of to-be-separated inequalities (4) is prohibitive, and the facet-defining subset of (4) is unknown [3]. Thus, we propose in Sec. 4 two primal feasible heuristics for the LMP.

### 3.3. Probabilistic Model

We now define a family of probability measures on lifted multicuts for which the maximally probable lifted multicuts are the solutions the LMP (Def. 1). This relates the coefficients  $c$  of the LMP to image and mesh data.

**Probability measures on multicuts.** Andres et al. [5] define, with respect to a graph  $G = (V, E)$  and with respect

to the Bayesian Network depicted in Fig. 1 (in black), a measure of the conditional probability of a  $y \in \{0, 1\}^E$ , given the feasible set  $Y_{E\emptyset}$  of the characteristic functions of all multicuts of  $G$  and given, for every edge  $e \in E$ , a vector  $x_e \in \mathbb{R}^n$  of  $n \in \mathbb{N}$  edge features. Specifically,

$$p_{\mathcal{Y}|\mathcal{X}, \mathcal{Y}} \propto p_{\mathcal{Y}|\mathcal{Y}} \cdot \prod_{e \in E} p_{\mathcal{Y}_e|\mathcal{X}_e} \quad (5)$$

$$\text{with } p_{\mathcal{Y}|\mathcal{Y}}(Y_{E\emptyset}, y) \propto \begin{cases} 1 & \text{if } y \in Y_{E\emptyset} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

They show that  $y$  maximizes  $p_{\mathcal{Y}|\mathcal{X}, \mathcal{Y}}$  if and only if it is a solution of the instance of the MP with respect to  $G$  and  $c \in \mathbb{R}^E$  such that

$$\forall e \in E : c_e = \log \frac{p_{\mathcal{Y}_e|\mathcal{X}_e}(0, x_e)}{p_{\mathcal{Y}_e|\mathcal{X}_e}(1, x_e)} \quad (7)$$

**Probability measures on lifted multicuts.** We extend the Bayesian Network of Andres et al. [5] in order to incorporate estimated probabilities not only for edges but also for pairs of nodes that are not neighbors.

The extension is depicted in Fig. 1 (in green). It contains one additional random variable  $\mathcal{Y}_f$  for every  $f \in F$ . The conditional probability measures  $p_{\mathcal{Y}|\mathcal{X}, \mathcal{Y}}$  consistent with the extended Bayesian Network have the form

$$p_{\mathcal{Y}|\mathcal{X}, \mathcal{Y}} \propto p_{\mathcal{Y}|\mathcal{Y}} \cdot \prod_{e \in E} p_{\mathcal{Y}_e|\mathcal{X}_e} \cdot \prod_{f \in F} p_{\mathcal{Y}_f|\mathcal{X}_E} \quad (8)$$

A realization of all random variables  $\mathcal{Y}$  is a 01-labeling  $y \in \{0, 1\}^{E \cup F}$  of all edges  $vw \in E \cup F$ . In order to constrain it to the characteristic functions of lifted multicuts, we consider (6) with  $Y_{EF}$  instead of  $Y_{E\emptyset}$ .

**Probabilistic Geodesic Lifting.** Estimating, for edges  $vw = e \in E$ , the probability  $p_{\mathcal{Y}_e|\mathcal{X}_e}$  of the nodes  $v$  and  $w$  being in distinct components, given features  $x_e$  defined by image and mesh data, is the classical problem of boundary estimation [8]. In our experiments described in Sec. 5, we build on recent work [20, 24, 33, 42] in this field.

Estimating, for pairs  $vw = f \in F$  of nodes  $v$  and  $w$  that are not neighbors, the probability  $p_{\mathcal{Y}_f|\mathcal{X}_E}$  of  $v$  and  $w$  being in distinct components is a much harder problem: As these nodes could be connected by any path in  $G$ , this probability depends on the features  $x_E$  of all edges. In our experiments, we define, for all  $vw = f \in F$ :

$$p_{\mathcal{Y}_f|\mathcal{X}_E}(0, x_E) := \max_{P \in vw\text{-paths}(G)} \prod_{e \in P} p_{\mathcal{Y}_e|\mathcal{X}_e}(0, x_e) \quad (9)$$

On the one hand, this under-estimates the probability as only one path is considered. On the other hand, it is the largest such under-estimate as a maximally probable such path is considered. Note also that  $-\log p_{\mathcal{Y}_f|\mathcal{X}_E}(0, x_E)$  can be computed efficiently using, e.g., Dijkstra's algorithm.

---

**Algorithm 1:** Greedy Additive Edge Contraction (GAEC)

---

```
1 while  $\mathcal{E} \neq \emptyset$  do
2    $ab := \operatorname{argmax}_{a'b' \in \mathcal{E}} \chi_{a'b'}$ 
3   if  $\chi_{ab} < 0$  then
4     break
5   contract  $ab$  in  $\mathcal{G}$  and  $\mathcal{G}'$ 
6   foreach  $ab \neq ab' \in \mathcal{E}'$  do
7      $\chi_{ab'} := \chi_{ab'} + \chi_{bb'}$ 
```

---

## 4. Efficient Algorithms

We now introduce two efficient algorithms (primal feasible heuristics) which are applicable to the LMP (Def. 1) and the MP (the special case of the LMP for  $F = \emptyset$ ).

Alg. 1 is an adaptation of greedy agglomeration, more specifically, greedy additive edge contraction. It takes as input an instance of the LMP defined by  $G = (V, E)$ ,  $F$  and  $c$  (Def. 1) and constructs as output a decomposition of the graph  $G$ . Alg. 2 is an extension of the Kernighan-Lin Algorithm [29]. It takes as input an instance of the LMP and an initial decomposition of  $G$  and constructs as output a decomposition of  $G$  whose lifted multicut has an objective value lower than or equal to that of the initial decomposition. Both algorithms maintain a decomposition of  $G$ , represented by graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  whose nodes  $a \in \mathcal{V}$  are components of  $G$  and whose edges  $ab \in \mathcal{E}$  connect any components  $a$  and  $b$  of  $G$  which are neighbors in  $G$ . Objective values are computed w.r.t. the larger graph  $G' = (V, E \cup F)$  and  $c$ .

### 4.1. Greedy Additive Edge Contraction

**Overview.** Alg. 1 starts from the decomposition into single nodes. In every iteration, a pair of neighboring components is joined for which the join decreases the objective value maximally. If no join strictly decreases the objective value, the algorithm terminates.

**Implementation.** Our implementation [1] uses *ordered adjacency lists* for the graph  $\mathcal{G}$  and for a graph  $\mathcal{G}' = (\mathcal{V}, \mathcal{E}')$  whose edges  $ab \in \mathcal{E}'$  connect any components  $a$  and  $b$  of  $G$  for which there is an edge  $vw \in E \cup F$  with  $v \in a$  and  $w \in b$ . It uses a *disjoint set data structure* for the partition of  $V$  and a *priority queue* for an ordered sequence of costs  $\chi : \mathcal{E} \rightarrow \mathbb{R}$  of feasible joins. Its worst-case time complexity  $O(|V|^2 \log |V|)$  is due to a sequence of at most  $|V|$  contractions, in each of which at most  $\deg \mathcal{G} \leq |V|$  edges are removed, each in time  $O(\log \deg \mathcal{G}') \in O(\log |V|)$ .

### 4.2. Kernighan-Lin Algorithm with Joins

**Overview.** Alg. 2 starts from an initial decomposition provided as input. In each iteration, an attempt is made to improve the current decomposition by one of the following transformations: 1. moving nodes between two neighboring

---

**Algorithm 2:** Kernighan-Lin Algorithm with Joins (KLj)

---

```
1 repeat
2   foreach  $ab \in \mathcal{E}$  do
3     if has_changed( $a$ ) or has_changed( $b$ ) then
4       update_bipartition( $\mathcal{G}, a, b$ )
5   foreach  $a \in \mathcal{V}$  do
6     if has_changed( $a$ ) then
7       repeat
8         update_bipartition( $\mathcal{G}, a, \emptyset$ )
9       until no changes
10 until no changes
```

---

components, 2. moving nodes from one component to an additional, newly introduced component, 3. joining two neighboring components. The main operation “update\_bipartition” is described below. It takes as input the current decomposition and a pair  $ab \in \mathcal{E}$  of neighboring components of  $G$  and assesses Transformations 1 and 3 for this pair. Transformations 2 are assessed by executing “update\_bipartition” for each component and  $\emptyset$ .

The operation “update\_bipartition” constructs a sequence of elementary transformations of the components  $a$  and  $b$  and a  $k \in \mathbb{N}_0$  such that the first  $k$  elementary transformations in the sequence, carried out in order, decrease the objective value maximally. Each elementary transformation consists in either moving a node currently in the component  $a$  which currently has a neighbor in the component  $b$  from  $a$  to  $b$ , or in moving a node currently in the component  $b$  which currently has a neighbor in the component  $a$  from  $b$  to  $a$ . The sequence of elementary transformations is constructed greedily, always choosing one elementary transformation that decreases the objective function maximally. If either the first  $k$  elementary transformations together or a complete join of the components  $a$  and  $b$  strictly decreases the objective value, an optimal among these operations is carried out.

**Implementation.** Our implementation [1] of Alg. 2 tags components that are updated in order to avoid that a pair of components that is fixed under “update\_bipartition” is processed more than once. In the operation “update\_bipartition”, we maintain the set  $\Omega \subseteq E$  of edges of  $G$  that straddle the components  $a$  and  $b$ . A substantial complication in the case of an LMP ( $F \neq \emptyset$ ) arises from the fact that moving a node  $v \in V$  from a component  $a \subseteq V$  to a neighboring component  $b \subseteq V$  might leave the set  $a \setminus \{v\}$  disconnected. Keeping track of these cut-vertices by the Hopcroft-Tarjan Algorithm [22] turned out to be impractical due to excessive absolute runtime. Our implementation allows for elementary transformations that leave components disconnected; we even compute the difference to the objective value incorrectly in such a case while constructing the sequence of elementary transformations. However, the first  $k$  elementary

transformations are carried out only if the correct difference to the objective value (computed after the construction of the entire sequence) is optimal. Our implementation of “update\_bipartition” has the worst-case time complexity  $O(|a \cup b|(|\Omega| + \deg G'))$ . The number of outer iterations of Alg. 2 is not bounded here by a polynomial but is typically small (less than 20 for all experiments in Sec. 5).

## 5. Experiments

### 5.1. Image Decomposition

We now apply both formulations of the graph decomposition problem, the Minimum Cost Multicut Problem (MP) [17] and the Minimum Cost Lifted Multicut Problem (LMP) defined in Sec. 3, in conjunction with both algorithms defined in Sec. 4, GAEC and KLj, to the Image Decomposition Problem posed by the BSDS-500 benchmark [8].

For every test image, we define instances of the MP and the LMP as described in Sec. 3. For each of these, we compute a feasible solution, firstly, by greedy additive edge contraction (GAEC, Alg. 1) and, secondly, by applying the extended Kernighan-Lin Algorithm (KLj, Alg. 2) to the output of GAEC. All decompositions obtained in this way are compared to the man-made decompositions in the BSDS-500 benchmark in terms of boundary precision and recall (BPR) [8] and variation of information (VI) [34]. The VI is split into a distance due to false joins, plus a distance due to false cuts, as in [30]. Statistics for the entire BSDS-500 test set are shown in Tab. 1 and Fig. 2 and are discussed below, after a specification of the experimental setup.

**Setup.** For every image, instances of the MP are defined w.r.t.: 1. the pixel grid graph of the image, 2. for every edge in this graph, *i.e.*, for every pair of pixels that are 4-neighbors, the probability estimated in [20] of these pixels being in distinct components, 3. a prior probability  $p^*$  of neighboring pixels being in distinct components. We vary  $p^* \in \{0.05, 0.10, \dots, 0.95\}$ , constructing one instance of the MP for every image and every  $p^*$ . For each of these instance of the MP, three instances of the LMP are defined by Probabilistic Geodesic Lifting (Sec. 3.3), one for each  $d^* \in \{5, 10, 20\}$ . Each experiment described in this section is conducted using one Intel Xeon CPU E5-2680 operating at 2.70 GHz (no parallelization).

**Results.** It can be seen from Fig. 2 that the algorithms GAEC and KLj defined in Sec. 4 terminate in a time in the order of  $10^3$  seconds for every instance of the MP and LMP we define, more than an order of magnitude faster than the state of the art [13]. Note that we do use the most efficient algorithm of [13] which exploits the planarity of the pixel grid graph. A more detailed comparison of KLj with CGC [13] and the implementation of KL in [4, 25] in terms of objective value and runtime is depicted in Fig. 3. It can be seen from this figure that our implementation of KLj is faster

	Boundary		Volume	
	F-measure	Covering	RI	VI [34]
gPb-owt-ucm [8]	0.73	0.59	0.83	1.69
SE+MS+SH[20]+ucm	0.73	0.59	0.83	1.71
SE+multi+ucm [9]	0.75	0.61	0.83	1.57
SE+MP GAEC	0.71	0.50	0.80	2.36
SE+MP GAEC-KLj	0.71	0.50	0.80	2.36
SE+MP 1-KLj	0.71	0.49	0.80	2.41
SE+MP GAEC-CGC	0.71	0.50	0.80	2.23
SE+LMP10 GAEC	0.71	0.51	0.80	2.33
SE+LMP10 GAEC-KLj	0.73	0.58	0.82	1.76
SE+LMP10 1-KLj	0.73	0.58	0.82	1.76
SE+LMP20 GAEC	0.71	0.52	0.80	2.22
SE+LMP20 GAEC-KLj	0.73	0.58	0.82	1.74
SE+LMP20 1-KLj	0.73	0.57	0.82	1.75

Table 1. Written above are boundary and volume metrics measuring the distance between the man-made decompositions of the BSDS-500 benchmark [8] and the decompositions defined by multicuts (MP), lifted multicuts (LMP) and top-performing competing methods [8, 9, 20]. Parameters are fixed for the entire data set (ODS).

than the implementation in [4, 25] also by more than an order of magnitude. This improvement in runtime facilitates our study of the MP and LMP with respect to the pixel grid graphs of the images in the BSDS-500 benchmark.

It can also be seen from Fig. 2 that feasible solutions of the MP found by GAEC are not improved significantly by either of the local search heuristics KLj or CGC [13]. Compared to the man-made decompositions in the benchmark in terms of BPR and VI, feasible solutions of the MP found by GAEC, improved by either CGC or KLj, are significantly worse than the state of the art [9] for this benchmark (Tab. 1).

In contrast, feasible solutions of the LMP found by GAEC are improved effectively and efficiently by KLj. CGC is not practical for the larger, non-planar graphs of the instances of the LMP we define; the absolute runtime exceeds 48 hours for every image and  $p^* = 0.5$ . Compared to the man-made decompositions in the benchmark, feasible solutions of the LMP found by GAEC and improved by KLj are not significantly worse than the state of the art [9] for this benchmark. The effect of changing  $p^*$  is shown for the average over all test images in Fig. 2 and for one image in particular in Fig. 4. The best decompositions for this image as well as for all images on average are obtained for  $p^* = 0.5$ . The effect of changing  $d^*$  is shown for the average over all test images in Fig. 3 (on the left). It can be seen from this figure that increasing  $d^*$  from 5 to 10 improves results while further increasing  $d^*$  to 20 does not change results noticeably.

### 5.2. Mesh Decomposition

We now apply our formulations and algorithms without any changes to the Mesh Segmentation Problem [16]. For the LMP and Alg. 2, we obtain state-of-the-art results.

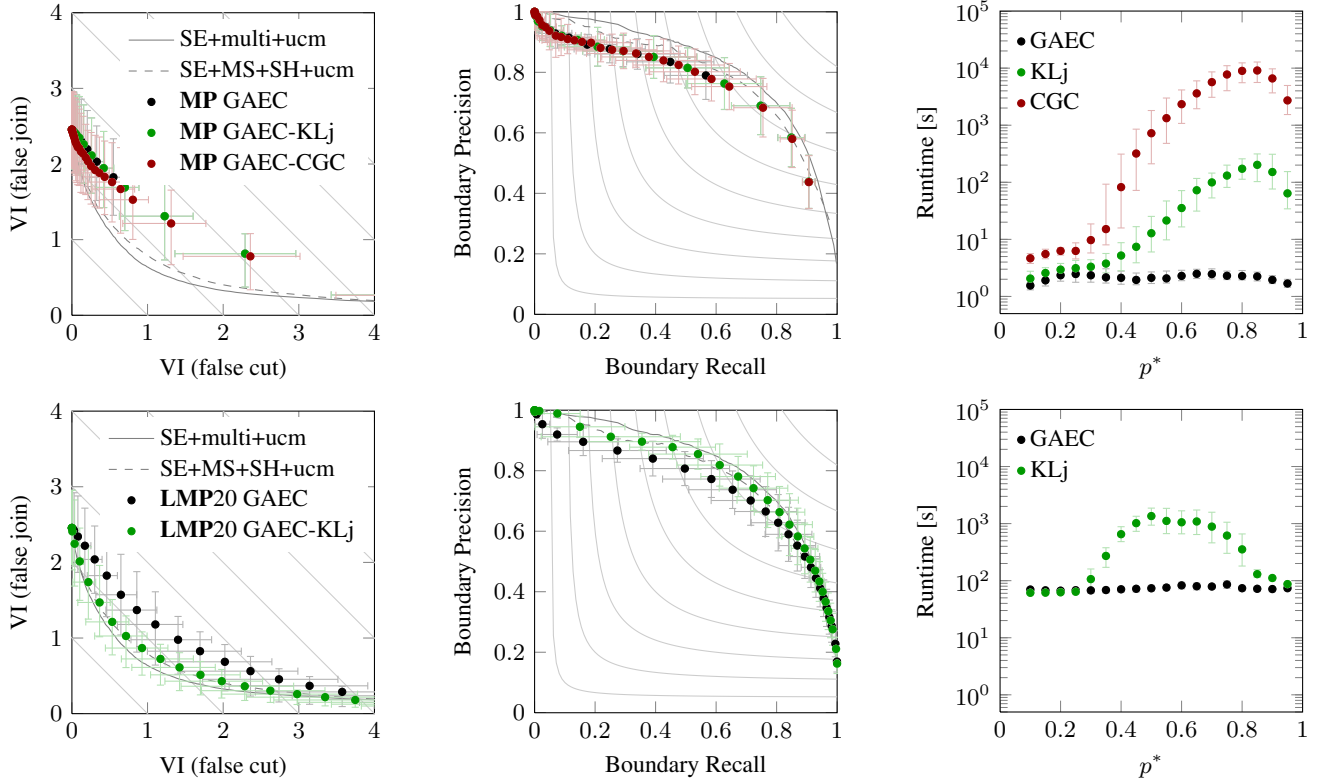


Figure 2. Depicted above is an assessment of the Multicut Problem (MP) and the Lifted Multicut Problem with  $d^* = 20$  (LMP20) in conjunction with Alg. 1 (GAEC) and Alg. 2 (KLj), in an application to the image decomposition problem posed by the BSDS-500 benchmark [8]. Every point in the figures above shows, for one problem and algorithm, the average over all test images in the benchmark. Depicted are, on the **left**, the variation of information (VI), split additively into a distance due to false cuts and a distance due to false joins, in the **middle**, the accuracy of boundary detection, split into recall and precision and, on the **right**, the absolute runtime. The state of the art SE+multi+ucm [9] and SE+MS+SH[20]+ucm are depicted as solid/dashed gray lines. Error bars depict the 0.25 and 0.75-quantile.

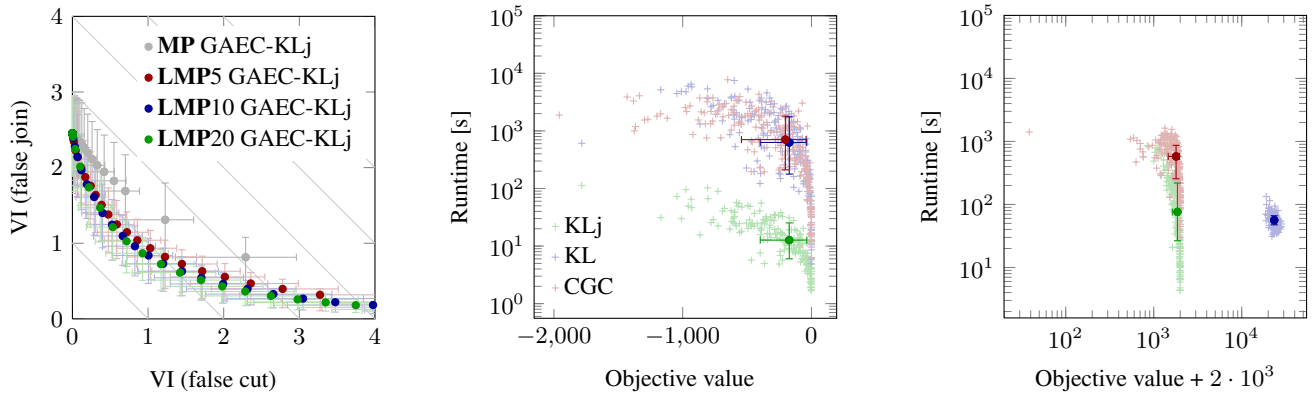


Figure 3. Depicted on the **left** is the effect of the lifting distance  $d^*$ . It can be seen that increasing  $d^*$  from 5 to 10 improves the quality of image decompositions as measured by the VI; further increasing  $d^*$  to 20 does not result in a measurable improvement. In the **middle**, a comparison of Alg. 2 (KLj) with CGC [13] and the implementation of the Kernighan-Lin Algorithm in [4, 25] (KL) is given. Every point corresponds to one instance of the MP ( $p^* = 0.5$ ) defined w.r.t. one test image in the BSDS-500 benchmark [8]. All algorithms are initialized here with the output of Alg. 1 (GAEC) for this instance. On the **right**, analogous results are shown for all algorithms initialized with a decomposition of the pixel grid into tiles of  $30 \cdot 30$  pixels. It can be seen that KLj strikes a favorable balance between objective value and runtime.

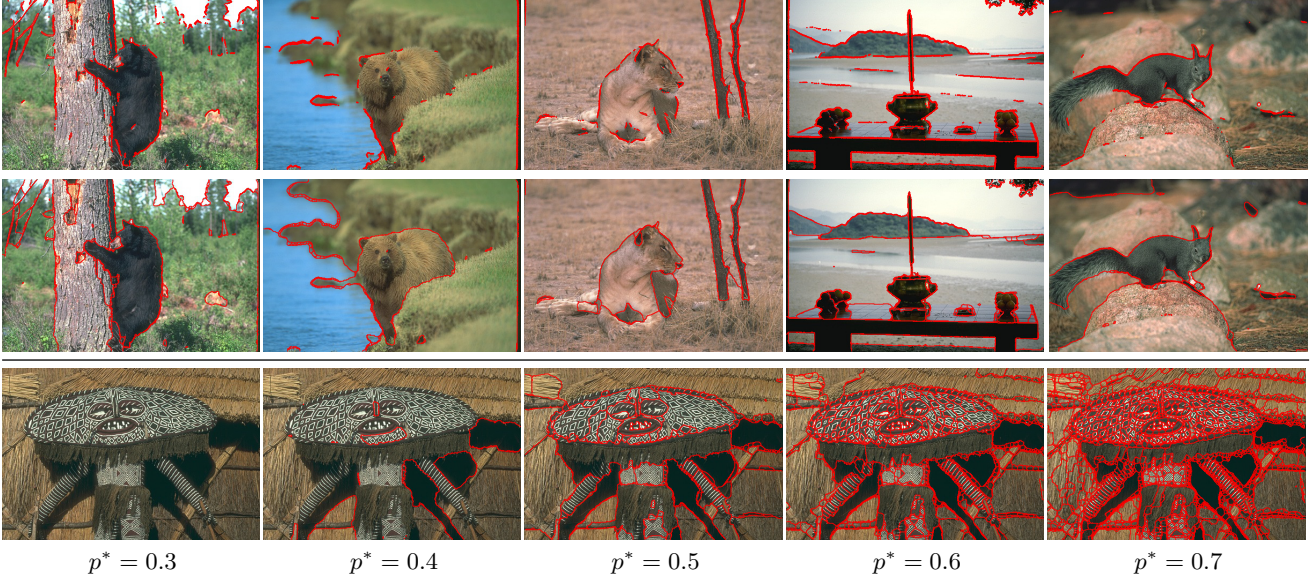


Figure 4. Depicted above is a comparison of decompositions defined by feasible solutions of the MP (**top** row for, the optimal  $p^* = 0.8$ ), with decompositions defined by feasible solutions of the LMP (**middle** row, for the optimal  $p^* = 0.5$  and  $d^* = 20$ ). All solutions are found by Alg. 2 (KLj), initialized with the output of Alg. 1 (GAEC). The decompositions defined by feasible solutions of the MP have closed contours but consist of tiny components on or near the boundary of desired components. This problem is overcome by feasible solutions of the LMP due to the long-range terms in the objective function. Depicted in the **bottom** row is the effect of the prior probability  $p^*$  which establishes a trade-off between over- and under-segmentation, reasonable in particular at  $p^* = 0.5$ .

	[42]	[33]	[24]		MP		LMP70		LMPopt	
	RI	RI	RI	VI	RI	VI	RI	VI	RI	VI
Human	0.89	0.88	0.88	1.43	0.21	2.93	0.86	1.62	0.87	1.79
Cup	0.80	0.79	0.90	0.35	0.74	0.68	0.89	0.39	0.90	0.39
Glasses	0.91	0.90	0.86	0.68	0.35	1.83	0.84	0.76	0.90	0.68
Airplane	0.89	0.87	0.92	0.67	0.69	1.39	0.92	0.82	0.92	0.83
Ant	0.98	0.96	0.98	0.37	0.93	0.67	0.98	0.42	0.98	0.42
Chair	0.89	0.88	0.95	0.43	0.79	0.98	0.93	0.55	0.93	0.55
Octopus	0.98	0.96	0.98	0.29	0.86	0.80	0.98	0.35	0.98	0.33
Table	0.90	0.94	0.94	0.28	0.76	0.81	0.94	0.28	0.94	0.29
Teddy	0.97	0.95	0.97	0.37	0.69	1.37	0.96	0.51	0.96	0.50
Hand	0.92	0.89	0.90	0.85	0.29	2.36	0.83	1.26	0.85	1.32
Plier	0.91	0.93	0.95	0.57	0.25	2.14	0.91	0.88	0.93	0.84
Fish	0.70	0.76	0.87	0.70	0.64	1.27	0.80	1.09	0.80	1.09
Bird	0.91	0.90	0.91	0.73	0.67	1.36	0.93	0.88	0.93	0.99
Armadillo	0.91	0.89	0.93	1.11	0.21	3.27	0.92	1.60	0.92	1.48
Bust	0.75	0.76	0.76	1.35	0.42	1.67	0.69	2.25	0.69	2.25
Mech	0.87	0.88	0.89	0.46	0.78	0.69	0.84	0.59	0.84	0.59
Bearing	0.83	0.82	0.91	0.45	0.87	0.60	0.84	0.69	0.84	0.69
Vase	0.88	0.83	0.85	0.75	0.55	1.34	0.83	0.90	0.84	0.87
FourLeg	0.86	0.82	0.86	1.34	0.30	2.58	0.84	1.84	0.84	1.72
Average	0.88	0.87	0.91	0.69	0.58	1.51	0.88	0.93	0.89	0.93

Table 2. Written above are boundary and volume metrics measuring the distance between the man-made decompositions of meshes in the Princeton Benchmark and the decompositions defined by multicuts (MP), lifted multicuts (LMP) and top-performing competing methods [42, 33, 24]. The evaluation is for a fixed parameter set for the entire database ( $p^* = 0.55$ ,  $d^* = 70$ : LMP70) as well as for the best parameter set we found for each class of meshes (LMPopt). Results for the MP are for the optimal  $p^* = 0.9$ .

**Setup.** The Princeton Segmentation Benchmark [16] consists of 19 classes, ranging from humans to man-made objects, each containing 20 meshes. Manual segmentations

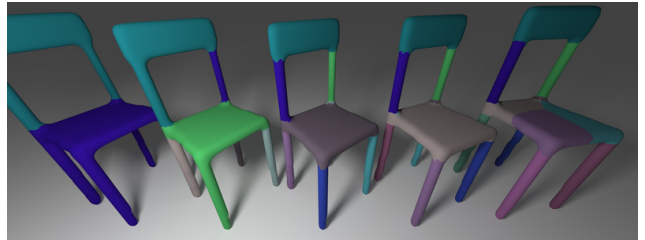


Figure 5. Depicted above is the effect of varying the prior probability  $p^*$  of adjacent triangles being in distinct components. Here,  $p^* \in \{0.5, 0.55, 0.58, 0.6, 0.62\}$ , from left to right.

of these meshes provide us with a ground truth for evaluation and supervised learning. We compute informative features known to provide good results in previous work [24, 14, 39]: curvatures (minimum, maximum, Gaussian and mean) computed at two different scales, shape diameter [36] and dihedral angle. Except the dihedral angle, which is computed for each edge, the curvatures and shape diameter are computed for each vertex of the mesh. To derive each of these criteria for an edge, we consider (1) its mean value over the two vertices of the edge, (2) its difference between the two vertices at each side of the edge and (3) the difference between its mean values computed on 1-ring neighborhoods at each side of the edge. This last combination provides additional robustness and multi-resolution behavior. This way, we obtain a 28-dimensional vector, which is more compact and efficient than the hundreds of features used in prior work [24, 39].

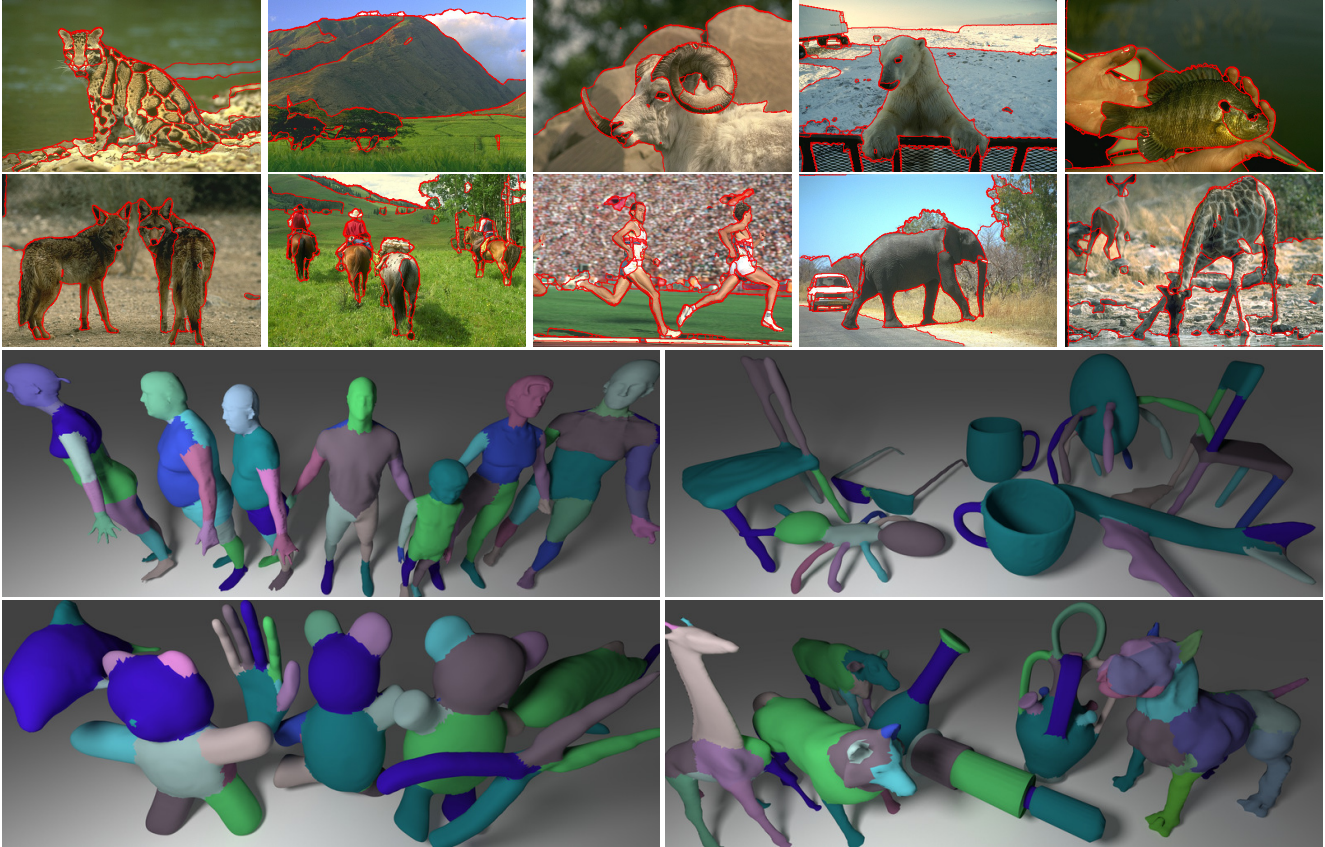


Figure 6. Depicted above is a sample of decompositions of images and meshes found by solving an instance of the Lifted Multicut Problem (LMP) using Alg. 2. Included are some cases where this approach fails.

Probabilities of edges being cut are learned from the ground truth using a Random Forest classifier, through a leave-one-out experiment similarly to previous work [24, 39]. We apply Alg. 2 on the dual graph of the mesh (one node per triangle), varying the prior probabilities  $p^*$  of neighboring triangles being in distinct components, and  $d^* \in \{60, 70, 80, 100\}$ .

**Results.** An evaluation in terms of Rand’s index (RI) and the VI is shown in Tab. 2. The results are slightly better than [42, 33] and close to those of [24]. However, Kalogerakis et al. [24] require a semantic labeling of the ground-truth, while our multicut formulation only requires boundary information. The median computation time per model is resp. 51 seconds for the lifting and 59 seconds for Alg. 2 on an Intel i7 Pentium laptop computer operating at 2.20 GHz. Graphs have a median of 18000 nodes and 27000 edges. A sample of our results is shown in Fig. 6. Varying the prior probability  $p^*$  of cuts allows for controlling the amount of over or under-segmentation, as shown in Fig. 5.

## 6. Conclusion

We have introduced a generalization of the Minimum Cost Multicut Problem (MP), the Minimum Cost Lifted Mul-

ticut Problem (LMP), which overcomes limitations of the MP in applications to image and mesh segmentation. We have defined and implemented two efficient algorithms (primal feasible heuristics) applicable to the MP and the LMP. We have assessed both algorithms in conjunction with both optimization problems in applications to image decomposition (BSDS-500 benchmark [8]) and mesh decomposition (Princeton benchmark [16]). In both applications, we have found solutions that do not differ significantly from the state of the art. This suggests that the LMP is a useful formulation of graph decomposition problems in vision.

**Acknowledgments.** M.K. and T.B. acknowledge funding by the ERC Starting Grant VideoLearn. N.B. and G.L. acknowledge support by the Agence Nationale de la Recherche (ANR) through the CrABEx project (ANR-13-CORD-0013).

## References

- [1] <http://www.andres.sc/graph.html>. 4
- [2] A. Alush and J. Goldberger. Ensemble segmentation using efficient integer linear programming. *TPAMI*, 34(10):1966–1977, 2012. 1
- [3] B. Andres. Lifting of multicuts. *CoRR*, abs/1503.03791, 2015. 3



- [4] B. Andres, T. Beier, and J. H. Kappes. OpenGM: A C++ library for discrete graphical models. *CoRR*, abs/1206.0111, 2012. 2, 5, 6
- [5] B. Andres, J. H. Kappes, T. Beier, U. Köthe, and F. A. Hamprecht. Probabilistic image segmentation with closedness constraints. In *ICCV*, 2011. 1, 3
- [6] B. Andres, T. Kröger, K. L. Briggman, W. Denk, N. Korogod, G. Knott, U. Köthe, and F. A. Hamprecht. Globally optimal closed-surface segmentation for connectomics. In *ECCV*, 2012. 1
- [7] B. Andres, J. Yarkony, B. S. Manjunath, S. Kirchhoff, E. Turetken, C. Fowlkes, and H. Pfister. Segmenting planar superpixel adjacency graphs w.r.t. non-planar superpixel affinity graphs. In *EMMCVPR*, 2013. 1, 3
- [8] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *TPAMI*, 33(5):898–916, 2011. 1, 2, 3, 5, 6, 8
- [9] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *CVPR*, 2014. 2, 5, 6
- [10] S. Bagon and M. Galun. Large scale correlation clustering optimization. *CoRR*, abs/1112.2903, 2011. 1, 2
- [11] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Machine Learning*, 56(1–3):89–113, 2004. 1, 2
- [12] T. Beier, F. A. Hamprecht, and J. H. Kappes. Fusion moves for correlation clustering. In *CVPR*, 2015. 1, 2
- [13] T. Beier, T. Kröger, J. H. Kappes, U. Köthe, and F. A. Hamprecht. Cut, Glue & Cut: A fast, approximate solver for multicut partitioning. In *CVPR*, 2014. 1, 2, 5, 6
- [14] H. Benhabiles, G. Lavoué, J.-P. Vandeborre, and M. Daoudi. Learning Boundary Edges for 3D-Mesh Segmentation. *Computer Graphics Forum*, 30(8):2170–2182, 2011. 2, 7
- [15] G. Bertasius, J. Shi, and L. Torresani. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In *CVPR*, 2015. 2
- [16] X. Chen, A. Golovinskiy, and T. Funkhouser. A benchmark for 3D mesh segmentation. *TOG*, 28(3):73, 2009. 1, 5, 7, 8
- [17] S. Chopra and M. Rao. The partition problem. *Mathematical Programming*, 59(1–3):87–115, 1993. 1, 2, 5
- [18] E. D. Demaine, D. Emanuel, A. Fiat, and N. Immerlica. Correlation clustering in general weighted graphs. *Theoretical Computer Science*, 361(2–3):172–187, 2006. 1, 2
- [19] M. M. Deza and M. Laurent. *Geometry of Cuts and Metrics*. Springer, 1997. 1, 2
- [20] P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *ICCV*, 2013. 2, 3, 5, 6
- [21] S. Hallman and C. Fowlkes. Oriented edge forests for boundary detection. In *CVPR*, 2015. 2
- [22] J. Hopcroft and R. Tarjan. Algorithm 447: Efficient algorithms for graph manipulation. *Commun. ACM*, 16(6):372–378, 1973. 4
- [23] P. Isola, D. Zoran, D. Krishnan, and E. H. Adelson. Crisp boundary detection using pointwise mutual information. In *ECCV*, 2014. 2
- [24] E. Kalogerakis and A. Hertzmann. Learning 3D mesh segmentation and labeling. *TOG*, 29(4):102, 2010. 2, 3, 7, 8
- [25] J. H. Kappes, B. Andres, F. A. Hamprecht, C. Schnörr, S. Nowozin, D. Batra, S. Kim, B. X. Kausler, T. Kröger, J. Lellmann, N. Komodakis, B. Savchynskyy, and C. Rother. A comparative study of modern inference techniques for structured discrete energy minimization problems. *IJCV*, 2015. 2, 5, 6
- [26] J. H. Kappes, M. Speth, B. Andres, G. Reinelt, and C. Schnörr. Globally optimal image partitioning by multicuts. In *EMMCVPR*, 2011. 1
- [27] J. H. Kappes, M. Speth, G. Reinelt, and C. Schnörr. Higher-order segmentation via multicuts. *CoRR*, abs/1305.6387, 2013. 1, 2
- [28] J. H. Kappes, P. Swoboda, B. Savchynskyy, T. Hazan, and C. Schnörr. Probabilistic correlation clustering and image partitioning using perturbed multicuts. In *SSVM*, 2015. 1
- [29] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell Systems Technical Journal*, 49:291–307, 1970. 1, 2, 4
- [30] M. Keuper, B. Andres, and T. Brox. Motion trajectory segmentation via minimum cost multicuts. In *ICCV*, 2015. 5
- [31] S. Kim, S. Nowozin, P. Kohli, and C. Yoo. Higher-order correlation clustering for image segmentation. In *NIPS*, 2011. 1, 2
- [32] S. Kim, C. Yoo, S. Nowozin, and P. Kohli. Image segmentation using higher-order correlation clustering. *TPAMI*, 36:1761–1774, 2014. 1, 2
- [33] O. Kin-Chung Au, Y. Zheng, M. Chen, P. Xu, and C.-L. Tai. Mesh Segmentation with Concavity-Aware Fields. *IEEE Transactions on Visualization and Computer Graphics*, 18(7):1125 – 1134, 2011. 2, 3, 7, 8
- [34] M. Meilă. Comparing clusterings—an information based distance. *J. Multivar. Anal.*, 98(5):873–895, 2007. 5
- [35] S. Nowozin and S. Jegelka. Solution stability in linear programming relaxations: Graph partitioning and unsupervised learning. In *ICML*, 2009. 1
- [36] L. Shapira, A. Shamir, and D. Cohen-Or. Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer*, 24(4):249–259, 2008. 7
- [37] J. Shi and J. Malik. Normalized cuts and image segmentation. *TPAMI*, 22(8):888–905, 2000. 1
- [38] P. Theologou, I. Pratikakis, and T. Theoharis. A comprehensive overview of methodologies and performance evaluation frameworks in 3D mesh segmentation. *CVIU*, 135:49–82, 2015. 2
- [39] Z. Xie, K. Xu, L. Liu, and Y. Xiong. 3D Shape Segmentation and Labeling via Extreme Learning Machine. *Computer Graphics Forum*, 33(5), 2014. 2, 7, 8
- [40] J. Yarkony, A. Ihler, and C. Fowlkes. Fast planar correlation clustering for image segmentation. In *ECCV*, 2012. 1
- [41] J. Yarkony, C. Zhang, and C. Fowlkes. Hierarchical planar correlation clustering for cell segmentation. In *EMMCVPR*, 2015. 1
- [42] J. Zhang, J. Zheng, C. Wu, and Jianfei Cai. Variational Mesh Decomposition. *TOG*, 31(3), 2012. 2, 3, 7, 8