

# Supplementary material for “Submodularity beyond submodular energies: coupling edges in graph cuts”

Stefanie Jegelka  
Max Planck Institutes Tübingen  
jegelka@tuebingen.mpg.de

Jeff Bilmes  
University of Washington  
bilmes@ee.washington.edu

This supplementary material contains details about the experiments, and further experimental results. On the theoretical side, this supplement contains the proof of Lemma 3 (approximation factor), and, finally, the relations to a number of other recent models in computer vision, summarized in Section 6 of the main paper.

## 1. Experiments

We begin with details about experiments, and present further qualitative and quantitative results that exceed the 8-page limit of the main paper.

### 1.1. Experimental setup

To investigate the effect of coupling edges, we compare cooperative cut (CoopCut) to the standard graph cut (GraphCut), and, for shrinking bias, also to curvature regularization.

To ensure equivalent conditions, all methods used the same weights on the terminal edges (*i.e.*, the same unary potentials), the same 8-neighbor graph structure, and the same inter-pixel edge weights. The unary potentials stem from color histograms [2], or from Gaussian mixture models (GMMs) with 5 components [12, 14]. The weight of an inter-pixel edge  $e = (v_i, v_j) \in \mathcal{E}_n$  is

$$w(e) = 2.5 + 47.5 \exp(-0.5 \|z_i - z_j\|^2 / \sigma). \quad (1)$$

Recall that  $z_i$  is the observed color vector for pixel (node)  $v_i$ . These weights are equivalent to those used in [14];  $\sigma$  is the variance of inter-pixel color differences).

The edge classes that define which edges cooperate in CoopCut are inferred by  $k$ -means clustering. If the image has shading, we use  $\ell_1$  distances between edge features  $\phi_r(e) = \log(z_j/z_i)$ , otherwise, we use squared Euclidean distances between linear features  $\phi_l(e) = z_j - z_i$ . In addition to the  $k$  classes, there is an extra class  $S'$  that contains all edges whose incident pixels have identical color, *i.e.*,  $\phi_r(e) = 0$  or  $\phi_l(e) = 0$ . This extra class did not grant any discount, *i.e.*,  $\vartheta_{S'} = 1$ . The other classes all shared the same  $\vartheta$ . In the main paper, we show results with a range of  $k$ , where  $k$  is the number of edge classes. If  $k = 1$ , then there is no structure in the cooperations, and discounts are granted uniformly. The difference in the results for  $k = 1$  and  $k > 1$  is due to the structure of cooperation, *i.e.*, the discounts being limited to groups of similar edges.

The curvature regularization was implemented as described in [5]. For all methods,  $\lambda$  refers to the “regularization coefficient”, the weight of the inter-pixel terms relative to the weight of the unary terms.

**Error.** The error was computed as the number of wrongly assigned pixels divided by the total number of unlabeled pixels (“unlabeled” refers to the user labels). The “joint error” was the weighted combination of total error and twig error,

$$\text{err}_{\text{joint}} = 2\text{err}_{\text{tot}} + \text{err}_{\text{twig}} \quad (2)$$

Figure 1 shows some examples of ground truth labelings for total and twig error. The errors are with respect to a hand-segmented ground truth (obtained using Adobe Photoshop CS4 and CS5). The ground truth labelings contain a few gray pixels at the boundaries, wherever the exact boundary (at the pixel level) was not completely clear. These were ignored when computing the error. This style of labeling is similar to that of the Grabcut data. For the Grabcut data, we used the given ground truth labelings, and the provided “Lasso” labels.

**Data.** We test cooperative cut on three types of images: (i) shaded grayscale and color images; (ii) non-shaded color images with fine, delicate objects (“twigs and legs”); and (iii) the grabcut image segmentation data [12, 1]. For (i) and (ii),



Figure 1. Examples for ground-truth labelings that were created by hand in Photoshop. The rightmost image of the beetle shows an example that was used to compute twig error — the gray region is ignored when computing the error.



Figure 2. Examples for artificially shaded images. Unary potentials were computed on the shaded images and thus less informative than those from the original images.

we took high-resolution images using either a Canon 7D or a Canon 5DMkII DLSR camera. These images were hand-segmented, reduced in size, and user labels were added by hand. The data is publicly available at [addurlhere](#). The data for (i) consists of 8 grayscale and 7 color images, the data for (ii) of 17 color images.

In Experiment 1 in the main paper, we used naturally shaded images (as shown in Section 1.3), and, in addition, artificially shaded images. The artificial shading tests whether shade that varies locally with higher frequency affects the improvement achieved by cooperative cuts; such variation affects the explicit modelling of shading in regions of the image. To shade images, we selected images from the “twigs and legs” data used in Experiment 2. To create a shading, the pixel at location  $(x, y)$  in the image was multiplied by  $0.4(1 + \sin(2\pi y/\gamma))$ , and  $\gamma$  was varied across images. Figure 2 shows some examples.

## 1.2. Shrinking bias and the effect of the coefficient $\lambda$

All segmentation methods we test can be described as minimizing an objective

$$E(x) = \text{unary-terms}(x) + \lambda \text{smoothness-term}(x). \quad (3)$$

The smoothness term acts as a regularizer. We first investigate the regularization curve, *i.e.*, the effect of varying  $\lambda$ .

Figure 3 illustrates the behavior of graph cut and cooperative cut with respect to the regularization coefficient. The total error shrinks in both methods with  $\lambda$ . As  $\lambda$  increases, graph cut destroys many of the fine, elongated structures, resulting in a high “twig” error. This is called “shrinking bias”. In contrast, cooperative cut preserves the fine structures even with higher regularization, and both total and twig error are minimized *simultaneously*. In addition, the plot shows that cooperative cut is not overly sensitive to parameter choice. The results in Section 1.3, in particular Figure 10, illustrate the behavior indicated in Figure 3.

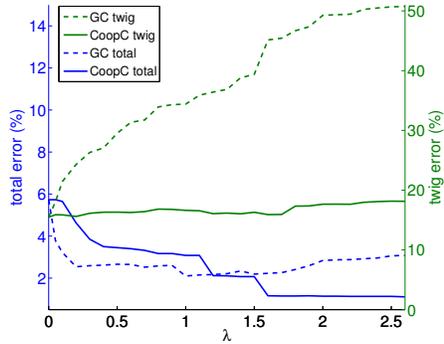


Figure 3. Behavior of the total and twig error with respect to  $\lambda$  for graph cut (dashed line) and cooperative cut with  $\vartheta = 10^{-4}$  (20 classes, solid line) on the “twigs and legs” data. The plot shows that a very low twig error, *i.e.*, preservation of elongated structures, coincides with low  $\lambda$  and high total error for graph cut. For cooperative cut, the twig error increases much more slowly with  $\lambda$ , so that both low total and low twig error are possible simultaneously. The plot also demonstrates that cooperative cut is not too sensitive to the choice of parameters.

The errors in Figure 3 are averages over the “twigs and legs” data, for  $\vartheta = 10^{-4}$  and 20 classes for CoopCut. Note that the twig error is in general much higher than the total error, because it refers to much fewer pixels.

### 1.3. Visual results

While Figure 3 indicates average behavior of the compared methods, here, we show what this behavior means visually (on high-resolution images). In general, there are two ways to optimize the parameters of a segmentation model. One approach is such that the parameters for a model are tuned to each image to give the best results — this is similar to the small number of sliders in a photo editing program that lets the user adjust the parameters to optimize the segmentation boundary of a given image. The visual results in the main paper were obtained in this way, meaning that we show the best parameter setting for each method. We saw that the best cooperative cut parameters were significantly better than the best graph cut parameters.

An alternative strategy is more appropriate for off-line use, where there might be many images that need to be segmented but there is not the option to adjust the parameters individually to each image. In this case, we must choose one set of parameters for each technique (e.g., graph cut or cooperative cut), and those parameters must be used for a range of images. In this section, we show results with such non-varying parameters, and we see that in this case as well, cooperative cut does much better.

#### 1.3.1 Shading in grayscale images (non-varying parameters)

Figures 4 and 5 show segmentations of shaded grayscale images obtained with fixed parameter settings, *i.e.*,  $\lambda = 0.1$  for graph cut, and  $(\lambda, 10^4\vartheta) = (6, 5)$  for cooperative cut. These are the parameters with the lowest average error. The error is indicated at the top of each image. Cooperative cut yields qualitatively and quantitatively better results.

#### 1.3.2 Shading in color images (non-varying parameters)

For the color images, Figure 6 shows example results that were all achieved with the same parameter setting. To see detailed differences, it is best to zoom in on a computer screen. The segmentation errors are indicated at the top of the images. For the plant, the difference is in the fine inner structures (holes) and inclusion of background at the bottom. For the calligraphy, cooperative cut preserves much more of the structure in the dark, and creates overall much cleaner boundaries. The parameters here are  $\lambda = 0.05$  for graph cut, and  $(\lambda, 10^4\vartheta) = (13, 4)$ .

A different shading example is shown in Figure 7, where the object’s shadow obstructs the segmentation.

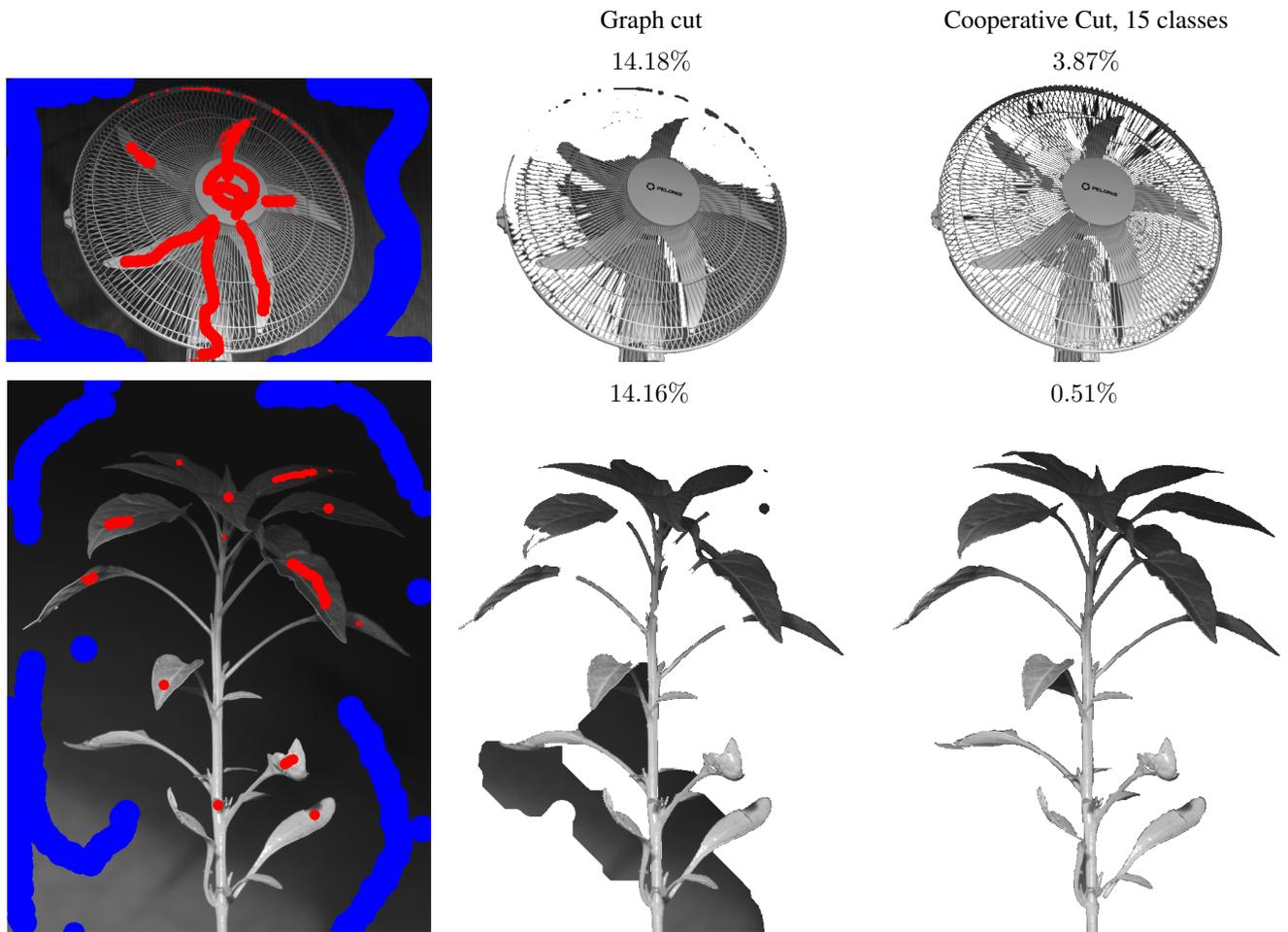


Figure 4. Results with fixed parameters for shading in grayscale images.

### 1.3.3 Elongated object parts in color images

Finally, Figures 8 and 9 present additional results for objects with fine, elongated structures. Such images were used in Experiment 2 in the main paper. The parameters are chosen for low error in each image, since otherwise the results with graph cut were too bad – Figure 10 shows sample results for minimum-error fixed parameters ( $\lambda = 1.5$  and  $\lambda = 0.05$  for graph cut, and  $(\lambda, 10^4\vartheta) = (1.5, 6)$  for CoopCut).

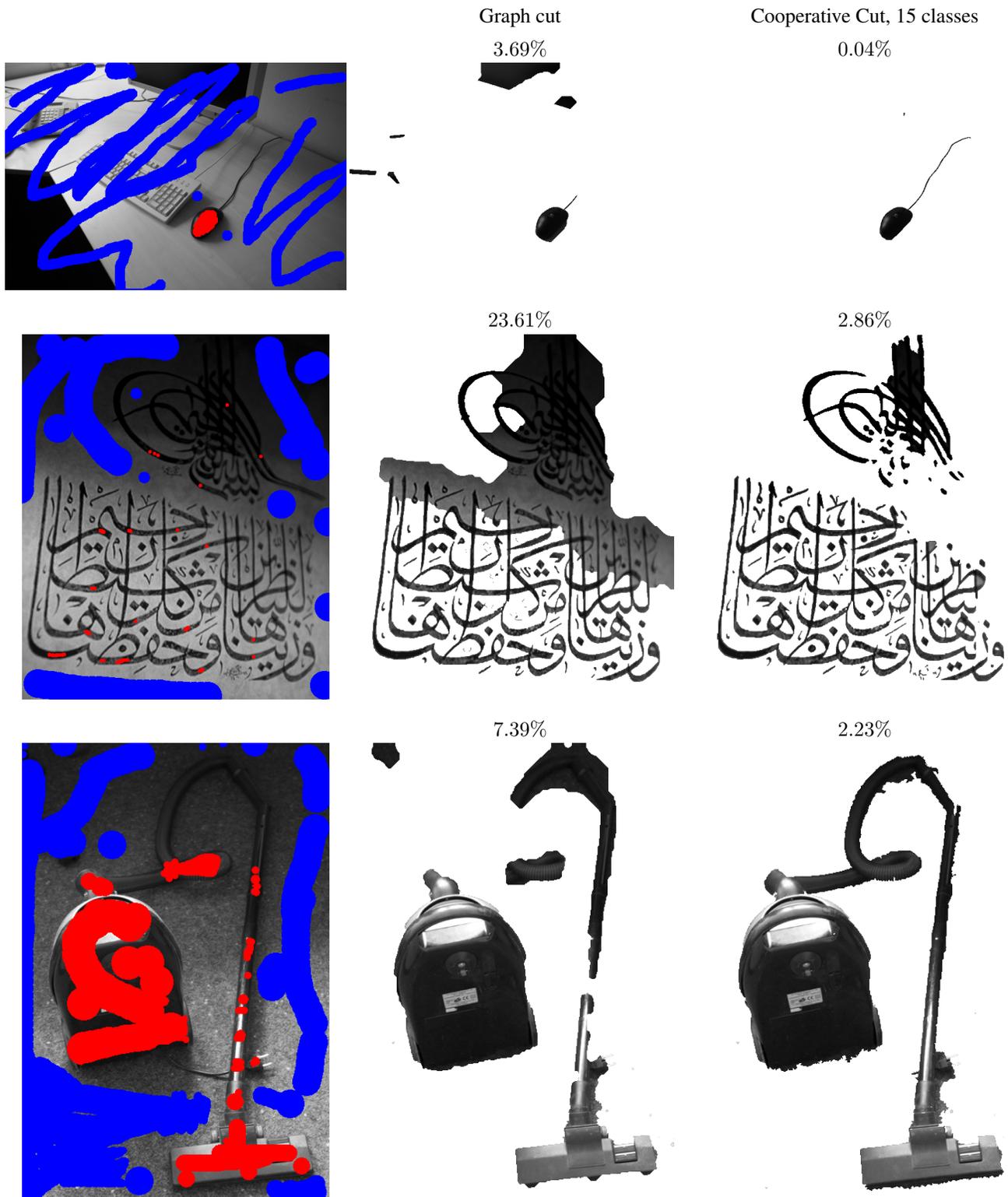


Figure 5. Results with fixed parameters for shading in grayscale images.

#### 1.4. Quantitative results

Tables 1, 2 and 3 present extended quantitative results for the experiments described in the main paper. For the errors reported here, the same parameters were used for all images in the data set.

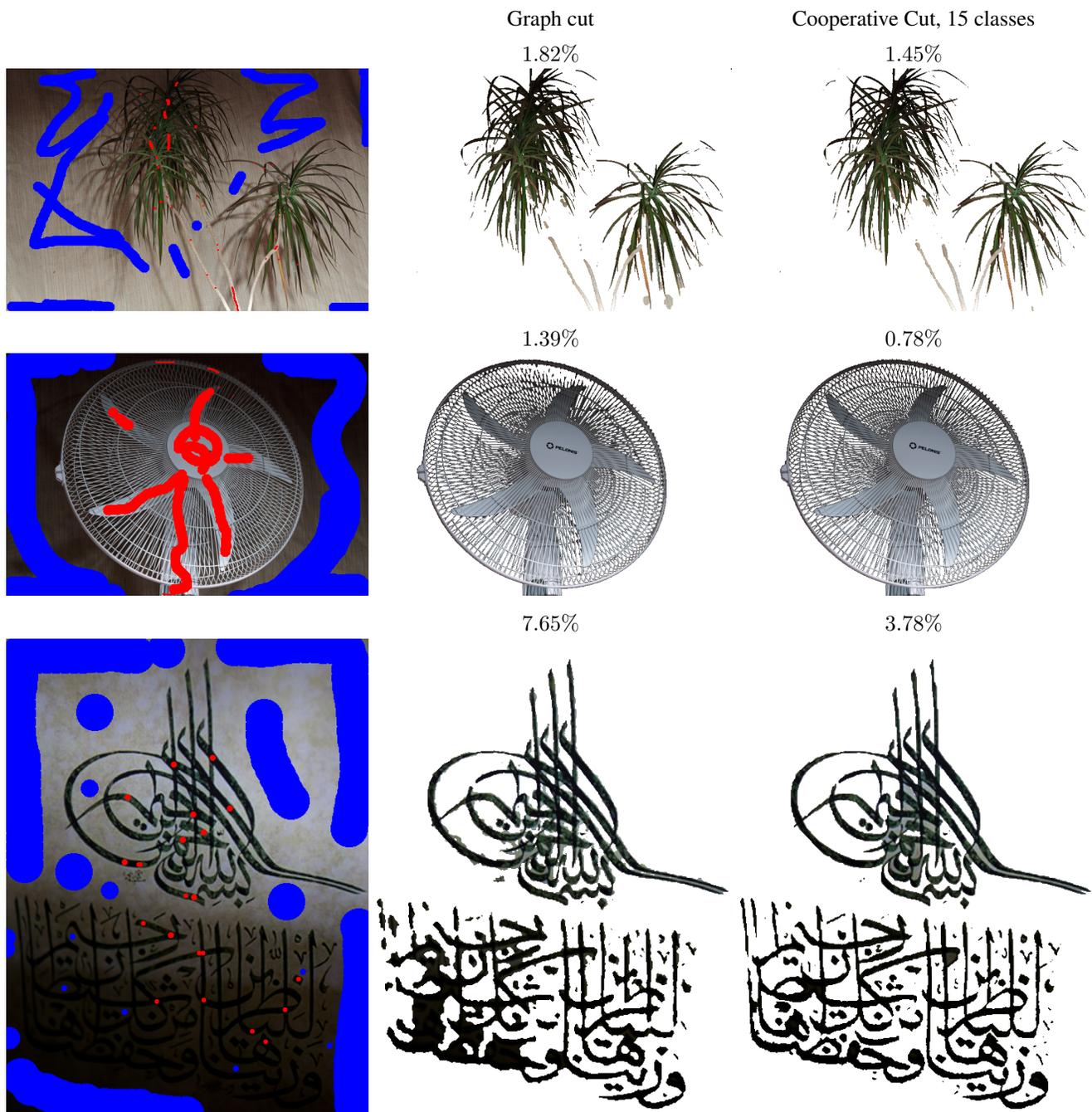


Figure 6. Results with fixed parameters for shading in color images.

As a baseline, we show not only the results for standard graph cut, but also the results for CoopCut with only one class (plus  $S'$ ), and for graph cut with logarithmic edge weights. The former shows the effect of indiscriminate coupling, *i.e.*, the discount is uniform on all edges (except  $S'$ ), and there is no group structure to it. The logarithmic weights are the non-cooperative equivalent of the edge features  $\phi_r$ . This baseline has the ratio information contained in  $\phi_r$ . CoopCut, however, always uses edge weights in Equation 1, and the ratios only for finding edge groups. The error denoted by “unary terms” is the error for graph cut with  $\lambda = 0$ , *i.e.*, the segmentation resulting from unary terms alone.

The results show that coupling edges improves on the results by graph cut, and a structured coupling via edge groups ( $k > 1$ ) improves on indiscriminate coupling. In fact, in many cases, a large part of the improvement is due to the grouping

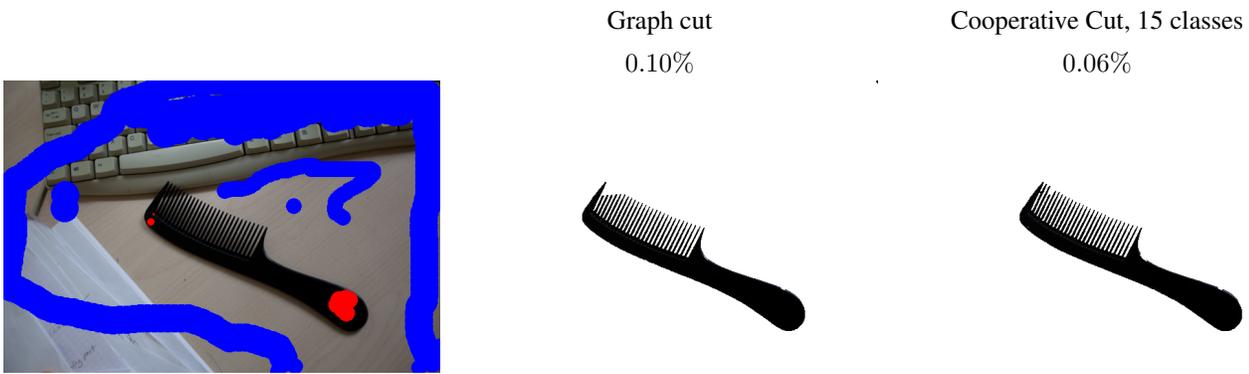


Figure 7. An example with a different type of shading. The comb's own shadow on the keyboard obstructs the segmentation of its teeth (GC  $\lambda = 0.2$ , CoopCut  $(\lambda, 10^4\vartheta) = (5.5, 10)$ )

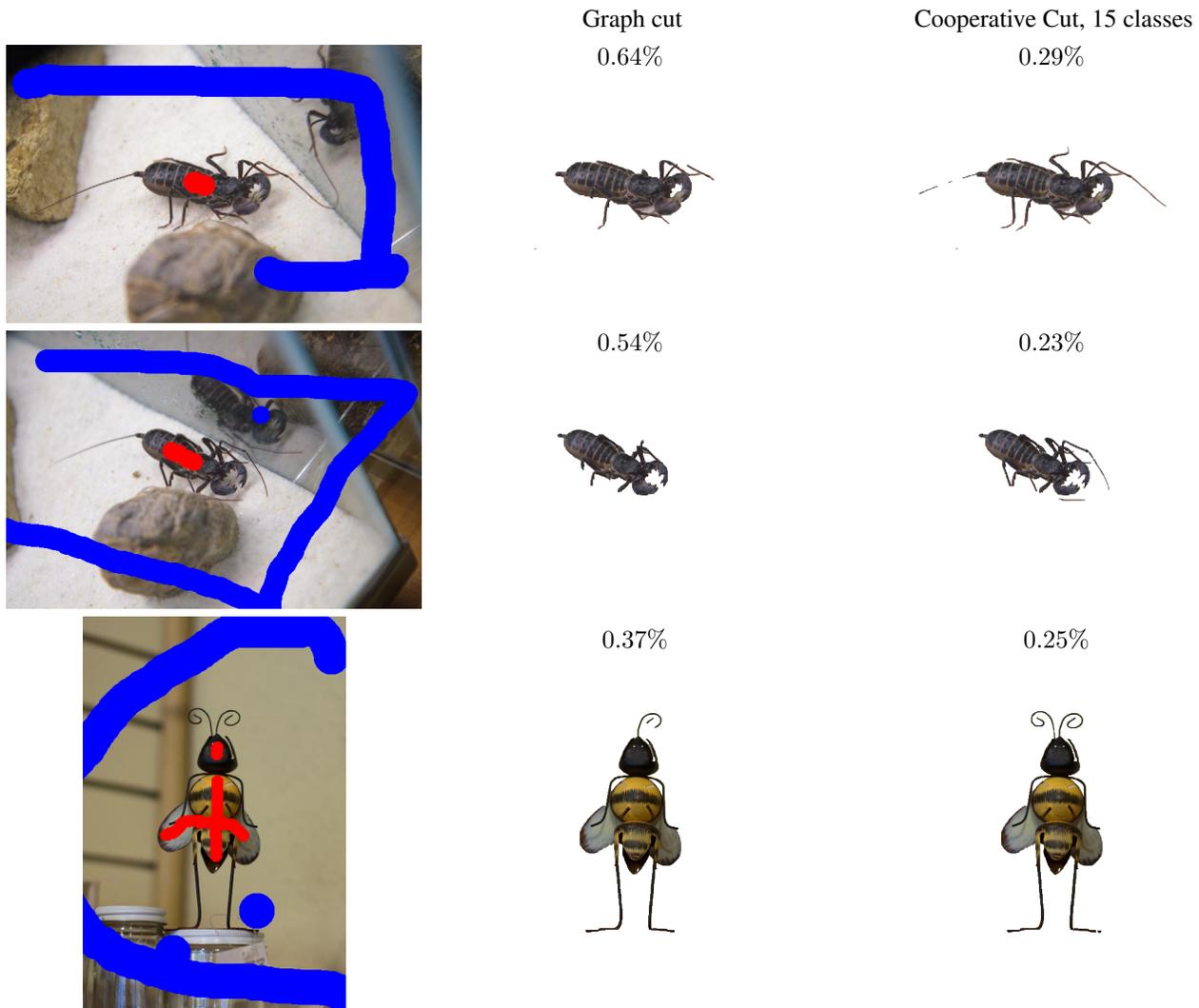


Figure 8. Additional results on “twigs and legs”, *i.e.*, objects with fine, elongated structures.

of edges. We do not know of any other method in the literature that uses such groupings.

Table 4 shows additional results on the Grabcut data, with larger numbers of edge classes. This experiment is a “sanity check”, since most objects in the Grabcut data set neither have delicate parts, nor are they shaded. They rather often have

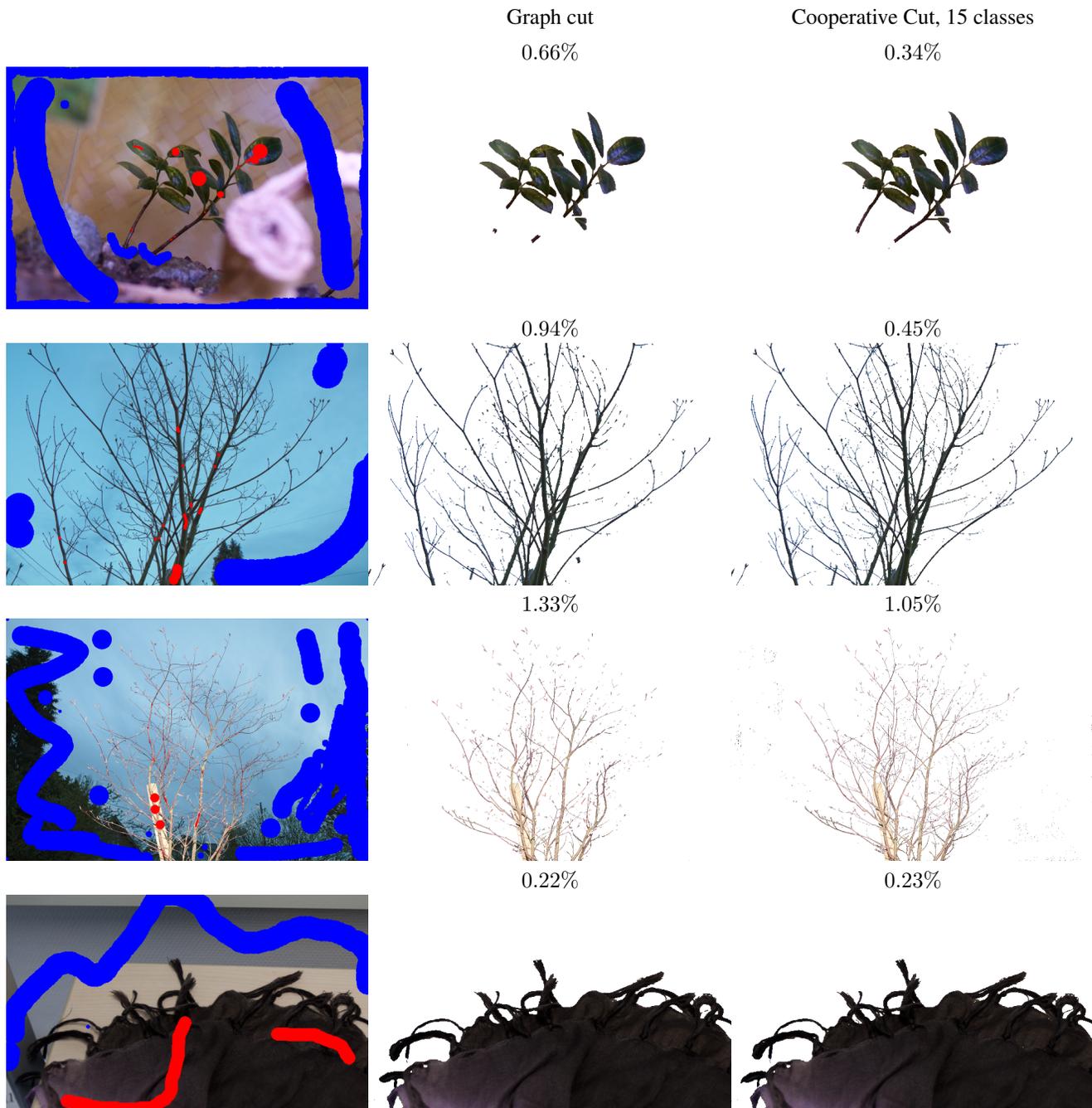


Figure 9. Additional results on “twigs and legs”, *i.e.*, objects with fine, elongated structures.

noisy backgrounds that demand regularization. Even in this case, CoopCut does not worsen the error compared to graph cut, it rather still improves on graph cut.

## 2. Recap: Notation and preliminaries

Before presenting the proofs and derivations mentioned in the main paper, we re-state our notation. The graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  is the structure underlying the potential  $E_f(\mathbf{x})$ . We denote a cut by  $\Gamma \subseteq \mathcal{E}$ . With cooperative cuts, the cost of a cut  $\Gamma$  is measured by a nonnegative, nondecreasing submodular function  $f : 2^{\mathcal{E}} \rightarrow \mathbb{R}_+$  defined on sets of edges ( $f$  replaces the usual sum of weights), *i.e.*,  $f(\Gamma)$ .

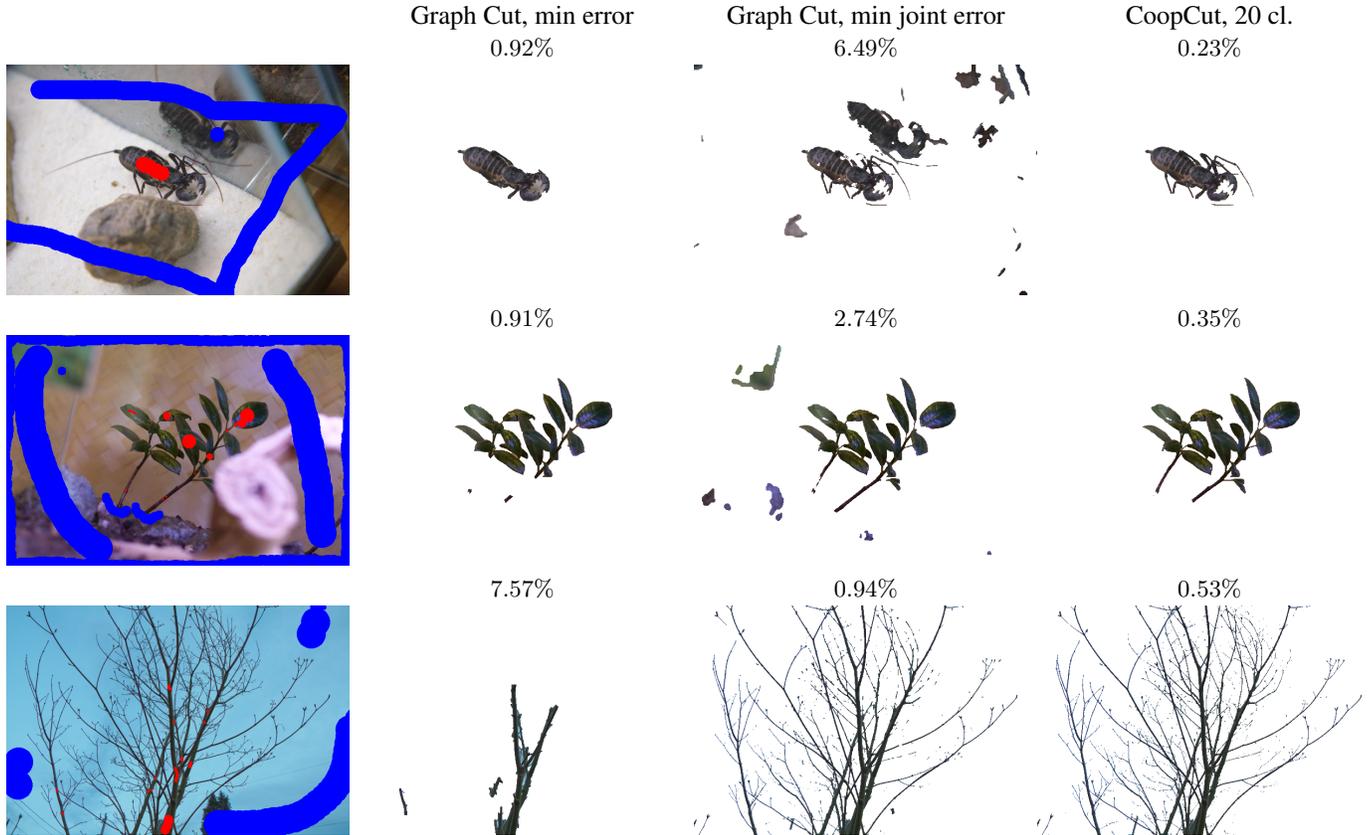


Figure 10. Effect of fixed parameters for graph cut and cooperative cut on the “twigs and legs” data. We show example results for parameters with minimum average error (graph cut, cooperative cut) and minimum average joint error (graph cut).

Natural shading: errors (in %)					Natural shading: parameters							
	grayscale		color		grayscale				color			
	GMM	hist	GMM	hist	GMM		hist		GMM		hist	
unary terms	15.66	17.42	4.42	8.18	$\lambda$	$10^4\vartheta$	$\lambda$	$10^4\vartheta$	$\lambda$	$10^4\vartheta$	$\lambda$	$10^4\vartheta$
GC	14.03	14.71	3.41	6.49	0.10	–	0.6	–	0.05	–	0.10	–
GC,log wts	13.67	14.13	3.63	6.54	2.60	–	1.9	–	0.05	–	0.05	–
CoopCut, 1	11.58	10.61	2.95	5.31	70.0	1	70.0	6	100.0	0	22.0	3
CoopCut, 10	4.39	5.02	1.67	3.05	8.0	5	19.0	5	15.0	2	5.7	8
CoopCut, 15	<b>3.63</b>	<b>4.27</b>	1.69	<b>2.94</b>	6.0	5	20.0	4	19.0	2	3.5	10
CoopCut, 20	4.33	4.48	<b>1.62</b>	3.00	7.0	5	21.0	3	14.0	3	6.5	4
curvature reg.	17.40	19.48	3.93	7.37	0.01	–	0.01	–	0.025	–	0.04	–

Table 1. Errors in percent (left) for the natural shading setting. Corresponding example visual results are in Sections 1.3.1 and 1.3.2. The right table displays the parameters with which the errors on the left were achieved.

Each variable has an associated node in  $\mathcal{V}$ . Using the bijection  $X(\mathbf{x}) = \{v_i \in \mathcal{V} | x_i = 1\}$  between sets  $X \subseteq \mathcal{V}$  and labelings  $\mathbf{x}$ , we consider potentials  $E_f(\mathbf{x}) = f(\Gamma(X(\mathbf{x})))$ . When we only refer to cuts in general in the sequel, then we will drop the arguments  $X(\mathbf{x})$ . Sets of nodes are denoted by  $X, Y \subseteq \mathcal{V}$ , and sets of edges by  $A, B, \Gamma \subseteq \mathcal{E}$ . We will also use the shorthand CoopCut for cooperative cut. As mentioned in the main paper, we consider directed graphs, because they allow to discriminate between the object and background side of the boundary, and this discrimination makes the similarity between boundaries more distinctive. That means, a boundary of a red object on green background is different from that of a green object on a red background. For more details on submodular functions, see [6], and for details on matroids, see [13].

High-frequ. shading (GMM): errors (in %)				
	min total error		min joint error	
	total	twig	total	twig
unary terms	5.50	14.55	5.50	14.55
GC	2.56	20.96	3.43	13.54
GC,log wts	2.58	23.21	4.11	13.52
CoopCut, 1	1.49	33.03	3.10	12.53
CoopCut, 10	1.26	14.79	1.65	12.47
CoopCut, 15	1.27	14.69	1.73	12.39
CoopCut, 20	1.29	18.10	1.62	12.01
CoopCut, 25	<b>0.78</b>	<b>13.34</b>	<b>1.57</b>	<b>8.38</b>
curvature reg.	3.38	34.50	4.70	14.08

High-frequ. shading (GMM): parameters				
	min total error		min joint error	
	$\lambda$	$10^4\vartheta$	$\lambda$	$10^4\vartheta$
GC	0.2	–	0.05	–
GC,log wts	0.2	–	0.02	–
CoopCut, 1	0.9	6	0.05	40
CoopCut, 10	0.6	12	2.40	2
CoopCut, 15	0.7	9	0.80	6
CoopCut, 20	10.0	1	5.00	1
CoopCut, 25	0.9	15	2.2	1
curvature reg.	0.15	–	0.01	–

Table 2. Errors and parameters for artificial, high-frequency shading. The discrimination between twig and total error shows that, while graph cut includes elongated structures at the cost of a higher total error, cooperative cut preserves them with much lower total error.

Twigs and legs (Expt. 2): errors (in %)								
	GMM				histograms			
	min total error		min joint error		min total error		min joint error	
	total	twig	total	twig	total	twig	total	twig
unary terms	5.73	15.47	5.73	15.47	10.49	21.46	10.49	21.46
GC	2.10	34.40	3.78	18.08	3.00	50.58	6.74	28.95
CoopCut, 1	1.25	34.35	4.73	15.60	2.85	48.11	8.71	21.89
CoopCut, 10	1.01	18.27	1.17	16.43	<b>1.74</b>	<b>33.18</b>	2.17	29.41
CoopCut, 15	1.01	26.32	1.02	16.36	1.76	37.59	<b>2.82</b>	<b>27.40</b>
CoopCut, 20	<b>0.98</b>	<b>17.78</b>	<b>1.16</b>	<b>15.91</b>	1.75	35.05	2.66	28.51
curvature	3.82	56.09	5.73	16.00	5.00	64.44	10.31	21.85

Twigs and legs: parameters								
	GMM				histograms			
	$\lambda$		$10^4\vartheta$		$\lambda$		$10^4\vartheta$	
	$\lambda$	$10^4\vartheta$	$\lambda$	$10^4\vartheta$	$\lambda$	$10^4\vartheta$	$\lambda$	$10^4\vartheta$
GC	1.0	–	0.05	–	1.3	–	1.6	–
CoopCut, 1	1.5	4	10.0	0	1.6	4	12.0	0
CoopCut, 10	1.5	7	2.6	1	3.2	5	1.3	7
CoopCut, 15	22.0	0	1.7	2	3.5	6	2.0	1
CoopCut, 20	1.8	5	1.6	1	3.0	6	2.8	1
curvature	0.3	–	0.001	–	0.5	–	0.002	–

Table 3. Errors and parameters for the “twigs and legs” data set used in Experiment 2. As on the high-frequency data set, cooperative cut better preserves fine structures at low total error, *i.e.*, without including too much background.

Grabcut data: errors and parameters							
	error (%)		parameters GMM		parameters hist.		
	GMM	hist	$\lambda$	$10^4\vartheta$	$\lambda$	$10^4\vartheta$	
GC	$5.33 \pm 3.7$	$6.88 \pm 5.0$	0.8	–	0.8	–	
CoopCut, $\phi_l$ , 10 cl.	$5.19 \pm 3.5$	$6.51 \pm 4.7$	0.8	10.0	30.0	2.0	
CoopCut, $\phi_l$ , 20 cl.	$4.95 \pm 3.2$	$6.27 \pm 4.2$	8.0	2.0	10.0	2.0	
CoopCut, $\phi_r$ , 10 cl.	$5.28 \pm 3.7$	$6.50 \pm 4.3$	0.5	30.0	10.0	2.0	
CoopCut, $\phi_r$ , 20 cl.	$4.79 \pm 3.1$	$6.12 \pm 4.0$	10.0	1.0	10.0	2.0	

Table 4. Results on the grabcut data with 10 and 20 classes and edge features  $\phi_l$  and  $\phi_r$ .

### 2.0.1 Induced submodular functions

We mention one property here that we will use in the sequel. Let  $\tilde{\mathcal{G}} = (\mathcal{V}_1, \mathcal{V}_2, \mathcal{F})$  be a bipartite graph, and let the neighborhood  $\mathcal{N}(X) = \{y \in \mathcal{V}_2 \mid (X \times \{y\}) \cap \mathcal{F} \neq \emptyset\}$  be the set of nodes in  $\mathcal{V}_2$  reachable from nodes  $X \subseteq \mathcal{V}_1$  via edges  $\mathcal{F}$ . Figure 11 illustrates such a neighborhood.

**Proposition 1.** *Let  $f_2 : 2^{\mathcal{V}_2} \rightarrow \mathbb{R}_+$  be a nondecreasing submodular function on subsets of  $\mathcal{V}_2$ . Then the induced function*

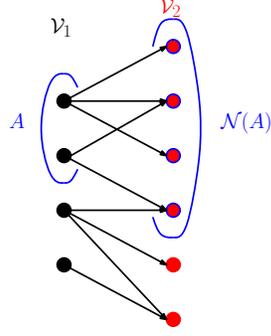


Figure 11. Bipartite graph and the neighborhood  $\mathcal{N}(A) \subseteq \mathcal{V}_2$  of set  $A \subseteq \mathcal{V}_1$ .

$$f_1 : 2^{\mathcal{V}_1} \rightarrow \mathbb{R}_+, \quad f_1(X) = f_2(\mathcal{N}(X)) \quad (4)$$

is nondecreasing submodular.

*Proof.* The proof relies on diminishing marginal costs (see also [13, §44.6g]). Let  $X \subseteq Y \subseteq \mathcal{V}_1 \setminus \{v\}$ . By the submodularity of  $f_2$ , it holds that

$$\begin{aligned} f_1(Y \cup \{v\}) - f_1(Y) &= f_2(\mathcal{N}(X) \cup \mathcal{N}(Y \setminus X) \cup (\mathcal{N}(v) \setminus \mathcal{N}(Y))) - f_2(\mathcal{N}(X) \cup \mathcal{N}(Y \setminus X)) \\ &\leq f_2(\mathcal{N}(X) \cup (\mathcal{N}(v) \setminus \mathcal{N}(Y))) - f_2(\mathcal{N}(X)) \\ &\leq f_2(\mathcal{N}(X) \cup \mathcal{N}(v)) - f_2(\mathcal{N}(X)) \\ &= f_1(X \cup \{v\}) - f_1(X). \end{aligned}$$

For the second inequality, we used that  $f_2$  is nondecreasing.  $\square$

Note that in the sequel, the nodes in the bipartite graph (that defines  $f_1$ ) can be the edges in the cooperative cut. Then  $f_1$  is a function on sets of edges in  $\mathcal{G}$ .

## 2.0.2 Move-making algorithms

A common technique for energy minimization with multiple labels are graph cut based move-making algorithms (see [3] for details), where the labels are changed iteratively in a restricted fashion. We will denote the initial labeling by  $\mathbf{x}'$  (with associated  $X'_k = \{v_i \mid x'_i = k\}$  for each label  $k \in \mathcal{L}$ ), and the labeling after the move by  $\mathbf{x}$  (with associated  $X_k = \{v_i \mid x_i = k\}$ ).

An  $\alpha\beta$  swap move changes the labels under the following restriction for fixed  $\alpha, \beta \in \mathcal{L}$ : (i) if  $x'_i \neq \alpha, \beta$ , then  $x_i = x'_i$ ; (ii) if  $x'_i \in \{\alpha, \beta\}$ , then  $x_i \in \{\alpha, \beta\}$ . That means, a label change can only occur from  $\alpha$  to  $\beta$  or vice versa. The swap move finds the minimum energy labeling in this restricted space of possible re-labelings.

An  $\alpha$  expansion move for a fixed  $\alpha \in \mathcal{L}$  may change labels as follows: (i) if  $x'_i = \alpha$ , then  $x_i = \alpha$ ; (ii) if  $x'_i = k \neq \alpha$ , then  $x_i \in \{k, \alpha\}$ . That means, a label can only be left alone or changed to  $\alpha$ , and  $X'_\alpha \subseteq X_\alpha$ . The  $\alpha$  expansion move finds the minimum-energy labeling amongst these re-labelings.

These definitions will be used in the sections below.

## 3. Approximation guarantee: Proof of Lemma 3

We begin the theoretical part by a proof of Lemma 3 in the emain paper. For more theoretical results, and empirical approximation factors, w refer the reader to [8].

As a reminder, we re-state Lemma 3 here:

**Lemma 3.** *Let  $\Gamma_\emptyset \in \operatorname{argmin}\{h_{\emptyset, f}(\Gamma) \mid \Gamma \subseteq \mathcal{E} \text{ an } (s, t)\text{-cut}\}$  be a minimum cut for  $h_{\emptyset, f}$ , and let  $\Gamma^* \in \operatorname{argmin}\{f(\Gamma) \mid \Gamma \subseteq \mathcal{E} \text{ an } (s, t)\text{-cut}\}$  be an optimal solution. Let  $\nu(\Gamma^*) = \min_{e \in \Gamma^*} \rho_e(\Gamma^* \setminus \{e\}) / \max_{e \in \Gamma^*} f(e)$ . Then*

$$f(\Gamma_\emptyset) \leq \frac{|\Gamma^*|}{1 + (|\Gamma^*| - 1)\nu(\Gamma^*)} f(\Gamma^*) \leq |\Gamma^*| f(\Gamma^*).$$

We note that for the graph we use in Section 5 of the main paper, an approximation bound of the same order can be derived for a convex relaxation followed by thresholded rounding. The algorithm is similar to that in [7], but uses two convex program formulations. We do not state details here since this is beyond the scope of this submission.

*Proof. (Lemma 3)* We bound  $f(\Gamma_\emptyset)$  from above. Let  $e' = \operatorname{argmax}_{e \in \Gamma^*} f(e)$ . The cost function  $f$  is subadditive, thanks to its submodularity and nonnegativity, and thus

$$f(\Gamma_\emptyset) \leq \sum_{e \in \Gamma_\emptyset} f(e) = h_{\emptyset, f}(\Gamma_\emptyset) \leq h_\emptyset(\Gamma^*). \quad (5)$$

The second inequality holds by the optimality of  $\Gamma_\emptyset$  for  $h_{\emptyset, f}$ . Now, again using subadditivity, we bound

$$h_{\emptyset, f}(\Gamma^*) \leq \sum_{e \in \Gamma^*} f(e) \leq |\Gamma^*| f(e'). \quad (6)$$

In consequence, it holds that  $f(\Gamma_\emptyset) \leq |\Gamma^*| f(e')$ .

Having this upper bound, we derive a lower bound on  $f(\Gamma^*)$ . As in the main paper, we denote the marginal cost by  $\rho_e(A) = f(A \cup \{e\}) - f(A)$ . Diminishing marginal costs imply that

$$\begin{aligned} f(\Gamma^*) &\geq f(e') + \sum_{e \in \Gamma^* \setminus \{e'\}} \rho_e(\Gamma^* \setminus \{e\}) \\ &\geq f(e') + (|\Gamma^*| - 1) \min_{e \in \Gamma^*} \rho_e(\Gamma^* \setminus \{e\}). \end{aligned} \quad (7)$$

Using the upper bound on  $f(\Gamma_\emptyset)$  and the lower bound on  $f(\Gamma^*)$ , we get

$$\frac{f(\Gamma_\emptyset)}{f(\Gamma^*)} \leq \frac{|\Gamma^*| f(e')}{f(e') + (|\Gamma^*| - 1) \min_{e \in \Gamma^*} \rho_e(\Gamma^* \setminus \{e\})}. \quad (8)$$

Dividing by  $f(e')$  yields the Lemma:

$$f(\Gamma_\emptyset) \leq \frac{|\Gamma^*|}{1 + (|\Gamma^*| - 1) \min_{e \in \Gamma^*} \rho_e(\Gamma^* \setminus \{e\}) / f(e')} \quad (9)$$

□

## 4. Other examples of cooperative cuts (Section 6)

Section 6 in the main paper lists some examples of other potentials that can be formulated as cooperative cuts. Some of these are special cases and thus allow for exact algorithms and specialized methods [9, 10, 4, 11]. In practice of course, a specialized exact algorithm is preferable if it performs well. On the other hand, cooperative cuts in their generality also perform very well, and the cooperative cut viewpoint is a conceptual enhancement that might allow natural extensions of the models discussed below to other cooperative cuts. New algorithms might also be inspired.

### 4.1. $P^n$ functions

Kohli *et al.* [9] introduce a general family of potential functions to which move-making algorithms can be applied. For a clique  $C$ , these potentials are of the form

$$\psi_C(\mathbf{x}_C) = g\left(\sum_{i, j \in C} \tilde{\psi}_C(x_i, x_j)\right), \quad (10)$$

where  $g : \mathbb{R} \rightarrow \mathbb{R}$  is concave non-decreasing and  $\tilde{\psi}_C$  is a symmetric pairwise potential satisfying  $\tilde{\psi}_C(a, b) \geq \tilde{\psi}_C(d, d)$  for all labels  $a, b, d \in \mathcal{L}$ . (This implies also that  $\tilde{\psi}_C(a, a) = \tilde{\psi}_C(b, b)$  even for  $a \neq b$ .) From their proofs it immediately follows that the inner sum of potentials is amenable to  $\alpha\beta$  swaps via graph cuts, and, if  $\tilde{\psi}$  is a metric, also to  $\alpha$  expansions [3] (take  $g$  as the identity function).

**Proposition 2.** *If the swap (expansion) move for the potential  $\sum_{i, j \in C} \tilde{\psi}_C(x_i, x_j)$  can be solved as a graph cut, then the swap (expansion) move for  $\psi_C$  can be solved as a Minimum Cooperative Cut.*

With standard graph cuts, the  $\alpha$  expansion is only possible for linear, increasing  $g$ .

*Proof.* Construct the graph to solve the move for the sum as a graph cut. Then, for any elementary cut  $\Gamma$  that can be reached by the move, it holds that  $w(\Gamma(X(\mathbf{x}_C))) = \sum_{i,j \in C} \tilde{\psi}_C(x_i, x_j) + c$  [3] for some constant  $c$  (this constant may be zero). Now, on this graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , define a submodular function  $f : 2^{\mathcal{E}} \rightarrow \mathbb{R}$ ,  $f(\Gamma) = g(c + \sum_{e \in \Gamma} w(e))$  where all edges cooperate. This function is submodular on  $2^{\mathcal{E}}$  because  $g$  is concave and nondecreasing. It follows that  $f(\Gamma(X(\mathbf{x}_C))) = g(w(\Gamma(X(\mathbf{x}_C)))) = \psi_C(\mathbf{x}_C)$ .  $\square$

As an illustration, we show what the graph and cut cost look like in the binary label case.

Let  $\nu_1 = \tilde{\psi}_C(1, 1)$ ,  $\nu_0 = \tilde{\psi}_C(0, 0)$  (with cooperative cuts,  $\nu_0 \neq \nu_1$  is possible), and  $\gamma = \tilde{\psi}_C(1, 0) \geq \nu_0, \nu_1$ . For each node  $v_i \in T$ , we add the terminal edges  $(s, v_i)$  and  $(v_i, t)$  with weights  $(|C| - 1)\nu_0$  and  $(|C| - 1)\nu_1$ , respectively. That means the terminal weights have one ‘‘unit’’ for each possible pairing of the variable. The non-terminal edges form a clique and have weight  $2\gamma - \nu_0 - \nu_1$ . Let, for this section,  $\mathcal{E}_t$  be the edges connected to  $t$ , and  $\mathcal{E}_s$  the edges connected to  $s$ . Then

$$f(\Gamma(X(\mathbf{x}_C))) = g((|C| - 1)(|\Gamma \cap \mathcal{E}_s| \nu_0 + |\Gamma \cap \mathcal{E}_t| \nu_1) + |\Gamma \cap \mathcal{E}_n|(2\gamma - \nu_1 - \nu_0)) \quad (11)$$

$$= g\left(\sum_{i \in C, x_i=0} \nu_0(|C| - 1) + \sum_{j \in C, x_j=1} \nu_1(|C| - 1) + \sum_{\substack{i,j \in C, \\ x_i=1, x_j=0}} (2\gamma - \nu_1 - \nu_0)\right) \quad (12)$$

$$= g\left(\sum_{\substack{i \in C, \\ x_i=1}} \sum_{\substack{j \neq i \in C, \\ x_j=1}} \nu_1 + \sum_{\substack{i \in C, \\ x_i=1}} \sum_{\substack{j \in C, \\ x_j=0}} (\nu_1 + 2\gamma - \nu_1 - \nu_0) + \sum_{\substack{i \in C, \\ x_i=0}} \sum_{\substack{j \neq i \in C, \\ x_j=0}} \nu_0 + \sum_{\substack{i \in C, \\ x_i=0}} \sum_{\substack{j \neq i \in C, \\ x_j=1}} \nu_0\right) \quad (13)$$

$$= g\left(\sum_{\substack{i \neq j \in C, \\ x_i=x_j=1}} \nu_1 + \sum_{\substack{i \neq j \in C, \\ x_i=x_j=0}} \nu_0 + \sum_{\substack{i,j \in C, \\ x_i > x_j}} (2\gamma - \nu_1 - \nu_0 + \nu_1 + \nu_0)\right) \quad (14)$$

$$= g\left(\sum_{\substack{i \neq j \in C, \\ x_i=x_j=1}} \nu_1 + \sum_{\substack{i \neq j \in C, \\ x_i=x_j=0}} \nu_0 + \sum_{\substack{i,j \in C, \\ x_i \neq x_j}} \gamma\right) \quad (15)$$

$$= \psi_C(\mathbf{x}_C). \quad (16)$$

## 4.2. $P^n$ Potts model

The  $P^n$  Potts potential is a generalization of the pairwise Potts potential and introduced in [9]. The  $P^n$  potential of a clique  $C$  is

$$\psi_C(\mathbf{x}) = \begin{cases} \gamma_k & \text{if } x_i = k \text{ for all } x_i \in C \\ \gamma_{\max} & \text{otherwise,} \end{cases}$$

for  $\gamma_{\max} > \gamma_k$ .

The  $\alpha\beta$  swap reduces to the binary case, since it is impossible to change the cost  $\gamma_{\max}$  if any variable in  $C$  has label  $x'_i \notin \{\alpha, \beta\}$ . For the cooperative cut, we construct a complete graph with nodes  $v_i$  for each  $x_i \in C$ . These intra-clique edges all have edge weight  $\gamma_{\max}$ . Then we add terminals  $s$  and  $t$ , with edges  $(s, v_i)$  and  $(v_i, t)$  for each node  $v_i$ . The  $s$ -edges get weight  $\gamma_\alpha$ , and the  $t$ -edges weight  $\gamma_\beta$ . As the cut cost function, we use the submodular function  $f : 2^{\mathcal{E}_t \cup \mathcal{E}_n} \rightarrow \mathbb{R}_+$ ,

$$f(\Gamma) = \max_{e \in \Gamma} w(e). \quad (17)$$

All nodes  $v_i$  for which  $(s, v_i)$  is cut by  $\Gamma$  will get label  $x_i = \alpha$ , all others  $x_j = \beta$ . If there are any  $v_i, v_j$  where  $\Gamma$  separates  $v_i$  from  $s$  (thus,  $x_i = \alpha$ ) and  $v_j$  from  $t$  (thus,  $x_j = \beta$ ), then  $\Gamma$  must also sever an edge between  $x_i$  and  $x_j$  with weight  $\gamma_{\max}$ , and hence  $f(\Gamma) = \gamma_{\max}$ .

The  $\alpha$  expansion is analogous. The only change is that the  $t$ -edges all get weight  $\gamma_\beta$  if initially, all nodes in the clique are labeled  $\beta$ , and  $\gamma_{\max}$  otherwise.

## 4.3. Robust $P^n$ potentials

Let  $N(\mathbf{x}_C)$  be the number of ‘‘deviating’’ labels in a clique  $C$ , i.e., the number of nodes taking the minority label, and let  $q \leq |C|/2$ . Then the robust  $P^n$  potential [10] is defined as

$$\psi_C(\mathbf{x}_C) = \begin{cases} N(\mathbf{x}_C)\gamma_{\max}/q & \text{if } N(\mathbf{x}_C) \leq q \\ \gamma_{\max} & \text{otherwise,} \end{cases}$$

Here, we model the binary version as a cooperative cut by letting terminal edges cooperate. We construct a graph  $\mathcal{G} = (\mathcal{V} \cup \{s, t\}, \mathcal{E}_t)$  with one  $v \in \mathcal{V}$  for each member of the clique. Each  $v$  is connected to the additional terminal nodes  $s, t$  by edges  $(s, v), (v, t)$ . Let  $S_1$  be the group of all edges  $(v, t)$  for  $v \in C$ , and  $S_2$  the group of all edges  $(s, v)$ , and

$$f_{S_i}(\Gamma) = \min\{|\Gamma \cap S_i|, q\} \gamma_{\max}/q \quad (18)$$

for  $i = 1, 2$ . This is a thresholded discount function as in Equation (13) in the main paper for edge weights  $\gamma_{\max}/q$ ,  $\theta_S = \gamma_{\max}$  and  $g(x) = 0$  and thus submodular. For the overall cut cost, we set

$$f(\Gamma) = f_{S_1}(\Gamma) + f_{S_2}(\Gamma), \quad (19)$$

and thus, it follows that

$$f(\Gamma(X(\mathbf{x}_C))) = f_{S_1}(\Gamma(X(\mathbf{x}_C))) + f_{S_2}(\Gamma(X(\mathbf{x}_C))) \quad (20)$$

$$= \min\{|\{i \mid x_i = 1\}|, q\} \gamma_{\max}/q + \min\{|\{i \mid x_i = 0\}|, q\} \gamma_{\max}/q \quad (21)$$

$$= \gamma_{\max} + \min\{N(\mathbf{x}_C), q\} \gamma_{\max}/q \quad (22)$$

$$= \psi_C(\mathbf{x}_C) + \gamma_{\max}. \quad (23)$$

Since  $\gamma_{\max}$  is constant, the potential  $\psi_C$  is equivalent to a cooperative cut.

#### 4.4. Co-occurrences of object labels

In class-based image segmentation, each pixel must be labeled by an object class and there are more than two classes. Ladický *et al.* [11] suggest a global potential  $f_L(L(\mathbf{x}))$  on the set of class labels  $L(\mathbf{x})$  used in the labeling  $\mathbf{x}$ .

Assume  $f_L : 2^{\mathcal{L}} \rightarrow \mathbb{R}_+$  is a nondecreasing and submodular function defined on sets of class labels. Then the  $\alpha$  expansion can be computed as a cooperative cut.

In the  $\alpha$  expansion, we can only re-label those  $x'_i$  that have a label different from  $\alpha$ . We create a graph with a node  $v_i$  for each  $x'_i \neq \alpha$  and a source and sink node. Figure 12 shows the construction. There, each label has a different color, and nodes are colored according to their initial labeling. We connect the source  $s$  to each  $v_i$  by an edge labeled  $\alpha$ , and we connect each  $v_i$  to  $t$  by an edge labeled  $x'_i$ . That is, we transfer the labels from nodes to edges. If the minimum cooperative cut severs edge  $(s, v_i)$ , then we set  $x_i = \alpha$ , otherwise edge  $(v_i, t)$  is cut and  $x_i$  retains the old label. An  $(s, t)$ -cut must cut one of those terminal edges. Since we will define a nondecreasing function, there is an optimal cut that cuts only one of the edges.

Let  $L_{\mathcal{E}}(\Gamma)$  be the set of labels of the edges in  $\Gamma \subseteq \mathcal{E}$ . If the current labeling  $\mathbf{x}'$  does not use the label  $\alpha$ , then we set the submodular cost function on edges to

$$f(\Gamma) = f_L(L_{\mathcal{E}}(\Gamma)). \quad (24)$$

This is a submodular function on subsets of edges. Submodularity follows from Proposition 1, if we let  $\mathcal{V}_1 = \mathcal{E}$  and  $\mathcal{V}_2 = \mathcal{L}$ , and if we connect each edge node in  $\mathcal{V}_1$  to its label in  $\mathcal{V}_2$ .

If the current labeling already uses the label  $\alpha$ , *i.e.*, there is an  $i$  with  $x'_i = \alpha$ , then we set the submodular cost function on edges to

$$f(\Gamma) = f_L(L_{\mathcal{E}}(\Gamma) \cup \{\alpha\}). \quad (25)$$

Submodularity follows as above. This construction is similar to the node-based construction in [11]. If  $\Gamma$  cuts edge  $(s, v_i)$ , then  $\alpha \in L_{\mathcal{E}}(\Gamma)$ , and we set  $x_i = \alpha$ . Otherwise,  $\Gamma$  cuts edge  $(v_i, t)$ , and then  $x_i = x'_i \in L_{\mathcal{E}}(\Gamma)$ . Thus, the cut cost is equivalent to the resulting labeling of nodes.

The construction for an  $\alpha\beta$  swap is very similar and uses only nodes that are initially labeled  $\alpha$  or  $\beta$  (see Figure 12).

In the main paper, we suggest the following submodular edge cost function. Let  $I_1, \dots, I_m$  be a set of training images, and  $L_I(I_k)$  the set of labels occurring in the labeling of image  $I_k$ . Let  $f_L : 2^{\mathcal{L}} \rightarrow \mathbb{R}_+$ ,

$$f_L(L) = \sum_{k=1}^M \delta(L \not\subseteq L_I(I_k)), \quad (26)$$

*i.e.*,  $f_L$  counts the number of images that do not contain all labels in  $L$ . This is a submodular function by Proposition 1: Let  $\mathcal{V}_1$  be all labels in  $\mathcal{L}$ , and  $\mathcal{V}_2$  be all images. Now, connect all labels to the images in which they do not occur. Then  $f_L(L)$  is the size of the neighborhood of  $L$  in the bipartite graph.

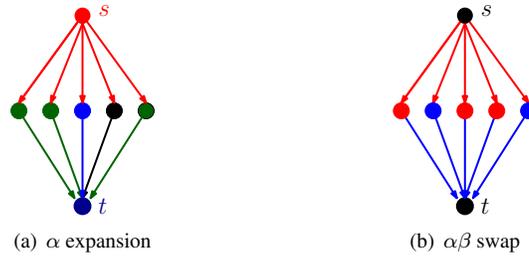


Figure 12. Graph construction for the  $\alpha$  expansion and  $\alpha\beta$  swap for class co-occurrence costs. The nodes  $v_i$  are aligned in the middle line. Here, the colors denote labels. Each node  $v_i$  is colored according to its initial label  $x_i$ . If edge  $(s, v_i)$  is cut, then we set  $x_i = \alpha$  (red label), otherwise we (a) keep the old label  $x_i'$  or (b) set  $x_i = \beta$  (blue label).

Similarly, the label cost part of energy  $(\star)$  in [4] is a submodular function on subsets of labels. Given label sets  $S \subseteq \mathcal{L}$ , and subset costs  $h_S \geq 0$ , the function is defined as

$$f_L(L) = \sum_S h_S \delta(L \cap S \neq \emptyset). \quad (27)$$

Submodularity again follows from Proposition 1, by setting  $\mathcal{V}_1 = \mathcal{L}$  and  $\mathcal{V}_2$  to the set of all  $S$  for which  $h_S > 0$ . The edges  $\mathcal{F}$  connect each label to the subsets that include it. Reference [4] also contains an extended discussion about optimization methods and applications of this energy, and relations to uncapacitated facility location. Note that the constructions with cooperative cut neither add additional nodes nor dense cliques to the graph; this might be an advantage for higher-order label interactions.

## References

- [1] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive GMMRF model. In *ECCV*, 2004.
- [2] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *ICCV*, 2001.
- [3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE TPAMI*, 23, 2001.
- [4] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov. Fast approximate energy minimization with label costs. *IJCV*, 2010. submitted.
- [5] N. El-Zehiry and L. Grady. Fast global optimization of curvature. In *CVPR*, 2010.
- [6] S. Fujishige. *Submodular Functions and Optimization*. Ann. of Discr. Math. Elsevier Science, 2nd edition, 2005.
- [7] S. Iwata and K. Nagano. Submodular function minimization under covering constraints. In *FOCS*, 2009.
- [8] S. Jegelka and J. Bilmes. Cooperative cuts: graph cuts with submodular edge weights. Technical Report TR-189, Max Planck Institute for Biological Cybernetics, 2010.
- [9] P. Kohli, M. Kumar, and P. Torr.  $P^3$  & beyond: Move making algorithms for solving higher order functions. *IEEE TPAMI*, pages 1645–1656, 2009.
- [10] P. Kohli, L. Ladický, and P. Torr. Robust higher order potentials for enforcing label consistency. *Int. J. Comp. Vision*, 82(3):302–324, 2009.
- [11] L. Ladický, C. Russell, P. Kohli, and P. Torr. Graph cut based inference with co-occurrence statistics. In *ECCV*, 2010.
- [12] C. Rother, V. Kolmogorov, and A. Blake. Grabcut – interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, 2004.
- [13] A. Schrijver. *Combinatorial Optimization*. Springer, 2004.
- [14] S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. In *CVPR*, 2008.