

Efficient Pixel-Grouping based on Dempster's Theory of Evidence for Image Segmentation

Björn Scheuermann¹, Markus Schlosser², Bodo Rosenhahn¹

¹ Leibniz Universität Hannover, Germany

{scheuermann, rosenhahn}@tnt.uni-hannover.de

² Technicolor Research & Innovation Hannover, Germany

markus.schlosser@technicolor.com

Abstract. In this paper we propose an algorithm for image segmentation using graph cuts which can be used to efficiently solve labeling problems on high resolution images or image sequences. The basic idea of our method is to group large homogeneous regions to one single variable. Therefore we combine the appearance and the task specific similarity with Dempster's theory of evidence to compute the basic belief that two pixels/groups will have the same label in the minimum energy state. Experiments on image and video segmentation show that our grouping leads to a significant speedup and memory reduction of the labeling problem. Thus large-scale labeling problems can be solved in an efficient manner with a low approximation loss.

1 Introduction

In the field of computer vision, discrete optimization using maximum flow algorithms has become very popular [1]. This has been driven by the fact that many problems such as image segmentation, stereo matching or shape matching are formulated using probabilistic models like Markov or conditional random fields (MRF or CRF respectively). The computation of the maximum a posteriori (MAP) solution for these models can be regarded as the discrete minimization of an energy function [2–4]. Many algorithms in literature are able to efficiently compute an approximate solution of the given optimization problem. Under some assumptions, e.g. such that the energy function is submodular, these methods are able to compute the exact minimum of the given energy function. We introduce an enhanced algorithm for grouping variables of the optimization problem which improves the general performance of maximum flow algorithms.

In parallel to the improvement of discrete energy minimization algorithms [3, 5, 6], the size of single images and image sequences increased significant. Compared to standard benchmark images, which have an approximate size of 120.000 pixels, nowadays commercial cameras capture images with many more pixels, e.g. up to 20 million. Since most energy functions for image segmentation or stereo matching contain one discrete variable per pixel, the minimization using maximum flow algorithms can be computationally extremely expensive. It has

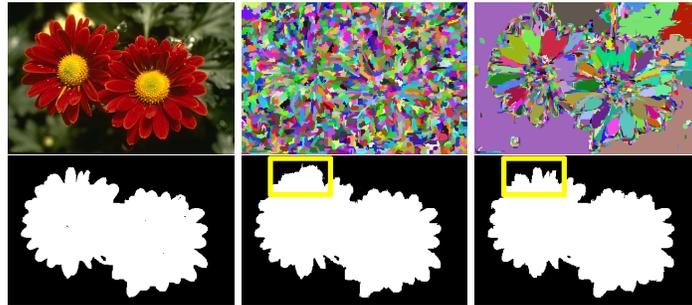


Fig. 1: Variable grouping for image segmentation. First row: original image; variable grouping of [9] with a budget of 1%; proposed variable grouping (COMPACTEDGE) with a budget of 1%; Second row: corresponding segmentation results. Using the same budget the proposed grouping is semantically more meaningful and leads to a smaller segmentation error.

been shown that the given algorithms are not applicable if the data of the problem does not fit into the physical memory [7, 8]. This observation has inspired researchers to develop more efficient energy minimization methods [5, 10, 11].

Related Work: Research on solving discrete optimization problems using maximum flow / minimum cut algorithms for applications in computer vision can be divided into the following approaches:

Augmenting paths: For computer vision problems, the most widely used algorithm is the Boykov and Kolmogorov augmenting paths algorithm [1, 12] (BK-algorithm). This algorithm efficiently solves moderately sized 2D and 3D problems with low connectivity.

Push-relabel: Most parallelized maximum flow / minimum cut algorithms are based on the push-relabel scheme [8]. For huge and highly connected grid graphs these methods outperform the traditional BK-algorithm [1]. In contrast to these methods, the proposed algorithm does not use special hardware to approximate the optimal solution.

Grouping of variables / graph sparsification: Besides the approaches to develop more efficient algorithms for the maximum flow / minimum cut problem, researchers are also trying to reduce the size of the labeling problem or the graph itself. One simple and widely used technique merges variables in the energy function into a smaller number of groups e.g. superpixels. Besides a number of well known image partitioning methods [13–16], Kim et al. presented a similar method [9] where the terms of the energy function and the algorithm proposed by Felzenszwalb and Huttenlocher [13] are used to decide if two variables should be merged. In [17] Scheuermann and Rosenhahn presented an algorithm for graph sparsification that does not change the optimal solution. The idea is to create a so called Slim Graph by merging nodes in the graph that do not change the maximum flow, meaning that these variables are guaranteed to have the same

label in the minimum energy state. Lermé et al. proposed a similar approach for graph sparsification by maintaining the maximum flow [18].

Multi-scale: Our work is also related to multi-scale methods for image labeling. The idea is to first solve the problem at low resolution using standard techniques [19–21]. This can be interpreted as a grouping of pixels into regular non-overlapping groups. The result of the low-resolution labeling is refined at the high-resolution in a following optimization step, where most variables of the problem are fixed.

Contribution: We propose an algorithm that merges variables of the energy function to small sets of non overlapping groups, so that each group can be represented by one single variable. The merging follows the idea of [13] and [9] where the grouping is based on appearance or the terms of the energy function respectively. In contrast to [9], we combine the task-specific similarity and the appearance using Dempster’s theory of evidence to compute the basic belief that two neighboring variables should be merged. Furthermore we do not directly penalize the size of a group by proposing new merging constraints (MAXEDGE and COMPACTEDGE), that follow our idea to allow large groups of variables in homogeneous regions. Instead of an accurate MAP our goal is to reduce to segmentation error. Therefore we use Dempster’s theory of evidence that is complementary to the terms of the energy function. We evaluate our method on standard benchmark images to show that our grouping achieves a better performance than the methods of [13] and [9]. Furthermore we quantify our algorithm on video sequences and high-resolution images to show that the segmentation, performed on top of our grouping, results in a similar segmentation with a dramatic reduction in computational costs and memory requirements.

Paper Organization: In Section 2 we continue with a review of discrete energy minimization, which is the basis for our segmentation framework, and recall the idea of Dempster’s theory of evidence. Section 3 introduces and explains the proposed grouping of variables. The details of our experiments and the analysis of the results are provided in Section 4. The paper finishes with a short conclusion.

2 Segmentation by Discrete Energy Minimization

The discrete energy $E : \mathcal{L}^n \rightarrow \mathbb{R}$ for the problem of binary image labeling addressed in this work can be written as the sum of unary φ_i and pairwise functions $\varphi_{i,j}$

$$E(x) = \sum_{i \in \mathcal{V}} \varphi_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \varphi_{i,j}(x_i, x_j), \quad (1)$$

where x is the labeling, \mathcal{V} corresponds to the set of all image pixels and \mathcal{E} is the set of all edges between pixels in a defined neighborhood \mathcal{N} . For the problem of binary image segmentation, the label set \mathcal{L} consists of a foreground (fg) and a background (bg) labels. The unary function φ_i is given as the negative log likelihood using a standard GMM model [6], defined as

$$\varphi_i(x_i) = -\log Pr(I_i | x_i = S), \quad (2)$$

where S is either fg or bg . The pairwise function $\varphi_{i,j}$ takes the form of a contrast sensitive Ising model, defined as

$$\varphi_{i,j}(x_i, x_j) = \gamma \cdot \text{dist}(i, j)^{-1} \cdot [x_i \neq x_j] \cdot \exp(-\beta \|I_i - I_j\|^2). \quad (3)$$

Here I_i and I_j describe the feature vectors of pixels i and j , e.g. RGB-colors. The parameter γ specifies the impact of the pairwise function. It has been shown that, using the defined unary and pairwise functions, the energy (1) is submodular and can hence be represented by a graph [12]. In this form, the global minimum of the energy can be computed with standard maximum flow algorithms [1].

To solve the labeling problem using maximum flow algorithms, the energy function needs to be represented by a graph. This can be done analogously to [12] by defining the graph $G = (\mathcal{V}_G, \mathcal{E}_G)$ as follows: the set of vertices is simply the set of pixels unified with two special vertices: $\mathcal{V}_G = \mathcal{V} \cup \{S, T\}$, where S denotes the source and T the sink. The set of edges consists of the set of all neighboring pixels plus edges between each pixel and the source and sink respectively: $\mathcal{E}_G = \mathcal{E} \cup \{(p, S), (p, T) \mid p \in \mathcal{V}\}$. The capacities $c(e)$ of each edge are defined analogously to Boykov et al. [12].

For the grouping of the variables, we follow the definitions given in [9] with our notation. A variable grouping of graph G is a graph $G' = (\mathcal{V}'_G, \mathcal{E}'_G)$ with energy function E' produced by a surjective map $m_G : \mathcal{V}_G \rightarrow \mathcal{V}'_G$ and the edge set $\mathcal{E}'_G = \{(s, t) \in \mathcal{V}'_G \times \mathcal{V}'_G \mid \exists (i, j) \in \mathcal{E}_G : m_G(i) = s \text{ and } m_G(j) = t\}$. Thus, the energy function for a variable grouping G' reads:

$$E'(x) = \sum_{i \in \mathcal{V}} \varphi_i(\hat{x}_{m_G(i)}) + \sum_{(i,j) \in \mathcal{E}} \varphi_{i,j}(\hat{x}_{m_G(i)}, \hat{x}_{m_G(j)}), \quad (4)$$

where \hat{x} is the labeling of the variable grouping. Solving this energy function on top of the grouping can be seen to correspond to the existing practice of using superpixels as a preprocessing step and defining the energy minimization problem on superpixels instead of pixels. Since most superpixels are directly derived from image properties, they perform poorly because the properties of the energy function, e.g. the unary term, are ignored. Figure 2 shows an example of a variable grouping and the corresponding graph based on the new energy function.

2.1 Dempster-Shafer Theory of Evidence

In this section we briefly review Dempster's theory of evidence, which is later used to define a similarity weight for two neighboring variables. The Dempster-Shafer theory of evidence, also called evidence theory, was introduced in the late 60s by A.P. Dempster [22], and more formally in 1976 by G. Shafer [23]. Later works [24, 25] applied it to image segmentation, and showed that it can be superior to Bayesian theory.

Evidence theory is a generalization of Bayesian theory which jointly represents inaccuracy and uncertainty information. The basic idea of the evidence

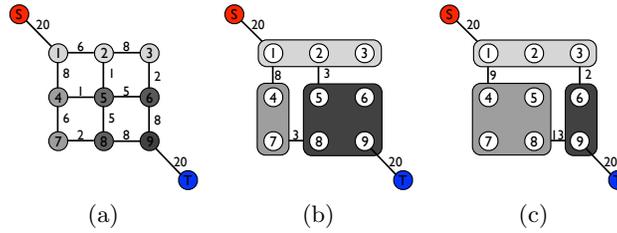


Fig. 2: Example variable grouping. The nodes from the original graph (a) are merged into three different groups of variables (b) and (c). The weights of the new graph are changed according to the new energy function. A good grouping (b) does not change the MAP solution of the original graph.

theory is to define a so-called mass function on a hypotheses set Ω . Let us note the hypotheses set Ω composed of n single mutually exclusive subsets Ω_i , symbolized by $\Omega = \{\Omega_1, \Omega_2, \dots, \Omega_n\}$. In order to express a degree of confidence for each element A of the power set $\wp(\Omega)$, an elementary mass function $m(A)$ is associated with it to indicate all confidences assigned to this proposition. The mass function m is defined by: $m : \wp(\Omega) \rightarrow [0, 1]$ and must fulfill the following conditions:

$$(i) \quad m(\emptyset) = 0 \quad (ii) \quad \sum_{A_n \subseteq \Omega} m(A_n) = 1. \quad (5)$$

The quantity $m(A)$ is interpreted as the belief strictly placed on hypothesis A . Compared to a Bayesian probability function, the mass function in evidence theory is the totality of belief. This belief is distributed on both simple and composed classes and models the impossibility to separate several hypotheses. Thereby the principal advantage of the evidence theory is characterized.

From the basic belief assignment m , a belief function $Bel : \wp(\Omega) \rightarrow [0, 1]$ can be defined as

$$Bel(A) = \sum_{A_n \subseteq A} m(A_n), \quad (6)$$

with $A_n \in \wp(\Omega)$. The belief function is the mass of hypothesis A plus the mass attached to all subsets of A . This can be interpreted as the total belief committed to a hypothesis. $Bel(A)$ is then the total positive effect the body of evidence has on a value being in A . It quantifies the minimal degree of belief of the hypothesis A .

A particular characteristic of Dempster-Shafer evidence theory differs from Bayesian theory: If $Bel(A) < 1$, then the remaining evidence $1 - Bel(A)$ does not need necessarily refute A (i.e. support its negation \bar{A}). That is, we do not have the so-called additivity rule $Bel(A) + Bel(\bar{A}) = 1$.

Dempster's rule of combination To unify evidence from a variety of features we use *Dempster's rule of combination*. This rule combines two independent

bodies of evidence, defined within the same frame of discernment, into one body of evidence. Let m_1 and m_2 be two mass functions associated to such independent bodies. Then the new body of evidence is defined by the mass function

$$m(A) = m_1(A) \otimes m_2(A) = \frac{\sum_{B \cap C = A} m_1(B)m_2(C)}{1 - \sum_{B \cap C = \emptyset} m_1(B)m_2(C)}. \quad (7)$$

Dempster's rule of combination computes a measure of agreement between two bodies of evidence and ignores the conflicting evidence through the normalization factor. Since Dempster's rule of combination is associative, we can combine information arising from more than two feature channels.

3 Dempster-Shafer based Variable Grouping

In this section we describe the details of our approach and show the similarities and differences to existing approaches. Let us assume a score function w measuring how similar two connected nodes are, such that small values indicate a strong similarity and large values dissimilarity. The idea of grouping nodes is as follows: (i) the first step is to sort all edges of the graph in ascending order so that edges with a small weight come first, (ii) for each edge in the list we merge nodes that fulfill a given constraint until we have sufficiently reduced the problem. The efficient graph-based segmentation method, proposed by Felzenszwalb and Huttenlocher [13], works exactly like this. To balance the size of a group and its internal coherence, a global criterion is used to decide if two groups can be merged. Algorithm 1 is identical to [13] and [9] using our notation. The merging constraint used in [13] and [9] is based on the so called internal difference

$$\text{Int}(C) = \max_{(i,j) \in \text{MST}(C,\mathcal{E})} w_{ij},$$

where $\text{MST}(C, \mathcal{E})$ is the minimum-weight spanning tree within the group C with a set of edges \mathcal{E} . $\text{Int}(C)$ is small if the nodes in group C are similar according to the defined edge weights. To decide whether two groups are merged, the algorithm compares the weight of the connecting edge between the two groups C_1 and C_2 and compares it with the internal difference $\text{Int}(C_i)$ of both groups. For our goal of grouping variables for energy minimization, this criterion makes sense since we want to build groups of variables that are similar and agree about their labeling. For the decision, [13, 9] use the function $\text{MInt}(C_1, C_2)$ defined as

$$\text{MInt}(C_1, C_2) = \min\{\text{Int}(C_1) + \tau(C_1), \text{Int}(C_2) + \tau(C_2)\},$$

where $\tau(C) = \frac{k}{|C|}$ penalizes the size of a group based on a free parameter k . According to Algorithm 1, when edge $w_{ij} \in \mathcal{E}_{\mathcal{G}}$ fulfills the equation

$$w_{ij} \leq \text{MInt}(C_i, C_j) \quad (8)$$

C_i and C_j are merged. As mentioned in [13], this graph based method is very efficient and easy to implement in $O(|\mathcal{E}_{\mathcal{G}}| \log |\mathcal{E}_{\mathcal{G}}|)$ time and memory.

Algorithm 1: Dempster-Shafer based Variable Grouping

```

1:  $(\mathcal{V}'_G, m) = \text{DempsterShaferGrouping}(G, \varphi, w)$ 
2: Input:
3:    $G = (\mathcal{V}_G, \mathcal{E}_G)$  // an instance of the graph
4:    $\varphi_i, \varphi_{i,j}$  // node and edge energies
5:    $w : \mathcal{E}_G \rightarrow \mathbb{R}$  // dissimilarity weights
6: Output:
7:    $\mathcal{V}'_G$  // set of grouped variables
8:    $m$  // surjective map
9: Algorithm:
10:   $\mathcal{V}'_G \leftarrow \mathcal{V}_G, \mathcal{E}'_G \leftarrow \mathcal{E}_G$ 
11:   $m \leftarrow \{(i, i) \mid i \in \mathcal{V}_G\}$ 
12:   $\pi \leftarrow \text{sort}(\mathcal{E}_G, w)$  {sort weights in ascending order}
13:  for  $e = 1, \dots, |\pi|$  do
14:     $(i, j) \leftarrow \pi_e$ 
15:    if  $m(i) = m(j)$  then
16:      continue {already merged}
17:    end if
18:    if  $w_{ij}$  fulfills given constraint then
19:      merge  $C_j$  and  $C_i$  in  $m, \mathcal{V}'_G$ 
20:    end if
21:  end for

```

3.1 Merging Function

The grouping resulting from the algorithms in [13] and [9] can be described as compact since the free parameter k in $\tau(C)$ penalizes the size of a group. In [9] the goal was to produce compact groups of variables that will have the same label according to the minimum energy state. Therefore the weight functions are based on the unary or pairwise potentials of the energy function. In contrast, our goal is to group as many variables as possible that are likely to have the same label according to the minimum energy state and to the ground truth labeling.

To allow big groups of variables, e.g. in homogeneous regions, we propose new merging constraints based on the maximum weight among outgoing edges. Instead of using a global criterion, balancing the size and the internal coherence of a group we merge all nodes that are connected by a sufficiently small edge. E.g. one could use the function $w_{ij} \leq W$ to merge all nodes connected by an edge smaller than the parameter W . As we will show in the experiments this simple constraint does not produce groups that agree with either the minimum energy state or the ground truth. To produce groups of homogeneous variables, we propose two new merging constraints based on the local edge weights of two nodes. The first constraint takes into account the maximum value of any edge connected to one of the two nodes. Therefore two components connected by the edge w_{ij} are grouped if

$$w_{max}(i, j) := \max \{w_{ik}, w_{lj} \mid (i, k), (l, j) \in \mathcal{E}_G\} \leq W_1 \quad (\text{MAXEDGE}). \quad (9)$$

This means that two nodes are merged if the weights of all edges adjacent to (i, j) , including the edge w_{ij} , are smaller than the parameter W_1 , which indicates that these nodes are somewhat similar. In our experiments the threshold W_1 is computed according to the distribution of the edge weights (66% of the edge weights are smaller than W_1). The idea of the proposed constraint is to have large groups of variables in all images regions except the borders of the objects. If a node (pixel) is near the border of an object there should be one edge with a high weight. With (9) this edge guarantees that the node is not merged with any neighbor. As a second constraint we also include the global criterion based on the minimum-weight spanning tree and the size dependent function τ , to balance the size of a group and its internal coherence, to allow somehow small compact groups of variables in regions that do not fulfill the MAXEDGE constrained, e.g. at the borders of an object. Thus, the decision is made according to

$$\text{MAXEDGE or } w_{ij} \leq \text{MInt}(C_i, C_j) \quad (\text{COMPACTEDGE}), \quad (10)$$

The differences of the proposed merging functions are discussed in the experiment section.

3.2 Weight Functions

We consider three classes of weight functions w_{ij} . The first two are well known weight functions that shall serve as comparison with the proposed one.

Felzenszwalb and Huttenlocher: In [13] Felzenszwalb and Huttenlocher take the pixel difference as the grouping weight. If I_i and I_j are the feature vectors of pixels i and j in the image, the weight is set to the norm of the difference:

$$w_{ij}^{FH} = \|I_i - I_j\|.$$

In our experiments on image segmentation, we will show that this method is not performing comparably, since the properties from the energy minimization problem are ignored.

Kim et al.: An approach very similar to [13] and ours, was proposed by Kim et al. in [9]. For comparison with the proposed method we use the defined UNARYDIFF weight function. In our experiments on standard benchmark images this weight function outperformed the others for the problem of binary image segmentation. The weight is defined as

$$w_{ij}^{ud} = \|\varphi_i - \varphi_j\|,$$

using the unary terms of the defined energy function. The weight describes the disagreement of the states between two variables and measure the task-specific similarity of two neighboring nodes.

Dempster-Shafer weighting function: Our proposed weight function includes the unary functions φ_i and φ_j and the pairwise terms φ_{ij} . Thereby we take into account the image information that are included in the pairwise function and the information included in the unary term, typically derived from a

discriminative classifier. Hence the proposed weight function can be seen as a combination of the two earlier presented ones which combines the image features with the task specific unary functions. To combine both informations we use Dempster’s theory of evidence. Therefore we define the weights based on the unary and pairwise functions

$$w_{ij}^{pairwise} = \varphi_{ij}(x_i, x_j) \quad \text{and} \quad w_{ij}^{unary} = \|\varphi_i - \varphi_j\|. \quad (11)$$

Since the co-domain of the weights are different, we normalize them individually to $[0, 1]$. That means for two variables with a similar feature vector $w_{ij}^{pairwise} \approx 1$. For w_{ij}^{unary} it means $w_{ij}^{unary} \approx 0$ if the negative log likelihood for two variables is similar for both states. Based on these weight functions, we define two mass functions over the hypothesis set $\Omega = \{\Omega_1, \Omega_2\}$, where Ω_1 means that the two variables are similar and Ω_2 that they are dissimilar:

$$\begin{aligned} m_1(\Omega_1) &= b_1 \cdot w_{ij}^{pairwise}, & m_1(\Omega_2) &= b_1 \cdot (1 - w_{ij}^{pairwise}), \\ m_1(\emptyset) &= 0, & m_1(\Omega) &= b_1, \\ m_2(\Omega_1) &= b_2 \cdot (1 - w_{ij}^{unary}), & m_2(\Omega_2) &= b_2 \cdot w_{ij}^{unary}, \\ m_2(\emptyset) &= 0, & m_2(\Omega) &= b_2, \end{aligned} \quad (12)$$

where b_i describes the belief we put on the different information sources. In all our experiments we equally weight the believe with $b_1 = b_2 = 0.5$. Now we fuse the two mass functions with Dempster’s rule of combination (7) and define the weights:

$$\begin{aligned} w_{ij}^{DS} &= 1 - Bel(\Omega_1) = 1 - m(\Omega_1) = 1 - m_1(\Omega_1) \otimes m_2(\Omega_1) \\ &= 1 - \left(\frac{m_1(\Omega_1) \cdot m_2(\Omega_1) + m_1(\Omega_1) \cdot m_2(\Omega) + m_1(\Omega) \cdot m_2(\Omega_1)}{1 - (m_1(\Omega_1) \cdot m_2(\Omega_2) + m_1(\Omega_2) \cdot m_2(\Omega_1))} \right) \end{aligned} \quad (13)$$

In contrast to [9], the proposed weight function allows for the combination with other information sources, such as the user initialization, the optical flow in video sequences, depth images or appearance information of an object.

4 Experiments

Our proposed grouping allows us to compute an approximate segmentation. Since the resulting graph for the energy minimization is much smaller the segmentation result differs from the original MAP solution. Since the goal of our grouping is a low segmentation error, and not an accurate MAP, we quantify our algorithm using three performance measures: (i) the segmentation quality with respect to the ground truth solution, (ii) the minimum segmentation error of a grouping, and (iii) the ratio of runtimes solving the MAP-problems (including the time for the grouping). Following, we describe the three measures in detail:

Segmentation error: The segmentation error is defined analogously to [2] as the ratio between the number of misclassified pixels and the number of pixels

Method	Avg. budget	Avg. $R_{mse}(x)$	Avg. $R_{se}(x)$
full MAP (reference)	100 (100)	0 (0)	0.075 (0.058)
FH-alg [13]	10.22 (10.22)	209.74 (209.74)	0.074 (0.063)
UNARYDIFF [9]	10.72 (10.84)	255.1 (219.08)	0.073 (0.065)
MAXEDGE	47.72 (15.21)	58.42 (4.21)	0.069 (0.058)
COMPACTEDGE	6.25 (5.00)	321.5 (63.52)	0.061 (0.058)

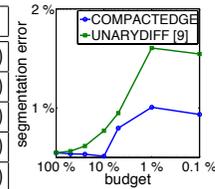


Table 1: Comparison of the proposed algorithm and two similar methods proposed in [13] and [9]. All values are averaged over 50 benchmark images using stroke (lasso) initializations. As can be seen our proposed method COMPACTEDGE performed best in terms of quality with a smaller budget. The proposed MAXEDGE has the lowest minimum segmentation error with the drawback of a bigger budget. The graph visualizes $R_{se}(x)$ for one image and different budgets.

in unclassified regions:

$$R_{se}(x) = \frac{\sum_{i \in \mathcal{V}_G} [x_i \neq x_i^{gt}]}{\text{no. pixels in unclassified regions}},$$

where x^{gt} is the ground truth labeling.

Minimum segmentation error: Another measure to quantify the quality of a grouping is given by the minimum segmentation error, that counts the minimum number of misclassified pixels by an optimal segmentation.

$$R_{mse}(x) = \sum_{i \in \mathcal{V}'_G} \min \left(\sum_{j \in m_{\mathcal{V}'_G}^{-1}(i)} [x_j^{gt} = fg], \sum_{j \in m_{\mathcal{V}'_G}^{-1}(i)} [x_j^{gt} = bg] \right).$$

Ratio of runtimes: To compute the ratio of runtimes we compare the time to compute the grouping and solve the reduced problem with the time solving the original problem.

We present an evaluation of the proposed method on small scale images of the Microsoft segmentation benchmark, used by Blake et al. [2]^{3,4} as well as on large scale images with up to 26 million pixels found on the web. For the problem of binary video segmentation we used video sequences from the KTH action dataset [26]⁵ and videos provided by Sand and Teller[27]⁶. In all experiments we use the same energy function proposed by Blake et al. [2] and the same set of parameters. The experiments were run on a MacBook Pro with 2.4 GHz Intel Core i5 processor and 4GB Ram. For all experiments we compare the proposed algorithm with the approaches of Felzenszwalb and Huttenlocher [13] and Kim et al. [9].

³ <http://research.microsoft.com/en-us/um/cambridge/projects/visionimagevideoediting/segmentation/grabcut.htm>

⁴ <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/>

⁵ <http://www.nada.kth.se/cvap/actions/>

⁶ <http://rvsn.csail.mit.edu/pv/>

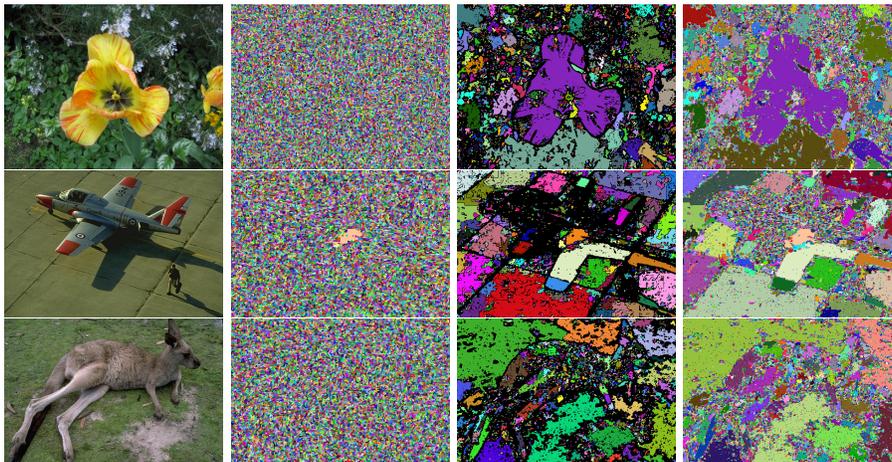


Fig. 3: Example for the different approaches for variable grouping. Columns: (i) original image; (ii) variable grouping using [9]; (iii) proposed method using MAXEDGE; (iv) proposed method using COMPACTEDGE; In contrast to [9] where the grouping produces superpixels that are comparable in size our proposed methods group large homogeneous regions to single variables.

Small-scale images: Table 1 shows the evaluation of the proposed algorithm on the Microsoft segmentation benchmark in comparison to the works of Felzenszwalb and Huttenlocher and Kim et al.. Since independently the benefit of the proposed weight w_{ij}^{DS} and the merging constraints is rather small, we only evaluate the combination that outperformed existing approaches.

We can observe that the combination of Dempster’s theory of evidence and the proposed constraint has a smaller average segmentation error with an even smaller budget. The small minimum segmentation error using the MAXEDGE constraint highlights that our idea to group large homogeneous regions to one single variable makes sense and the proposed weights based on Dempster’s theory of evidence reliably find those regions. In combination with small groups at the objects boundaries the proposed COMPACTEDGE constraint outperforms the existing approaches. See also Figure 3 for a visual comparison of the different approaches.

High-resolution images: To evaluate the segmentation quality and the possible speedup of the proposed method we used large-scale images with up to 20 MP and down sampled these images to several image-sizes. Similar to the experiments on small-scale images and video sequences the difference in segmentation quality is small and the reduction of runtime is dramatic for large images. As already shown by DeLong and Boykov [8] the BK-algorithm is inefficient and unusable if the graph does not fit into the physical memory. For those large MAP inference problems the ratio of runtime was approximately 0.08 using a budget

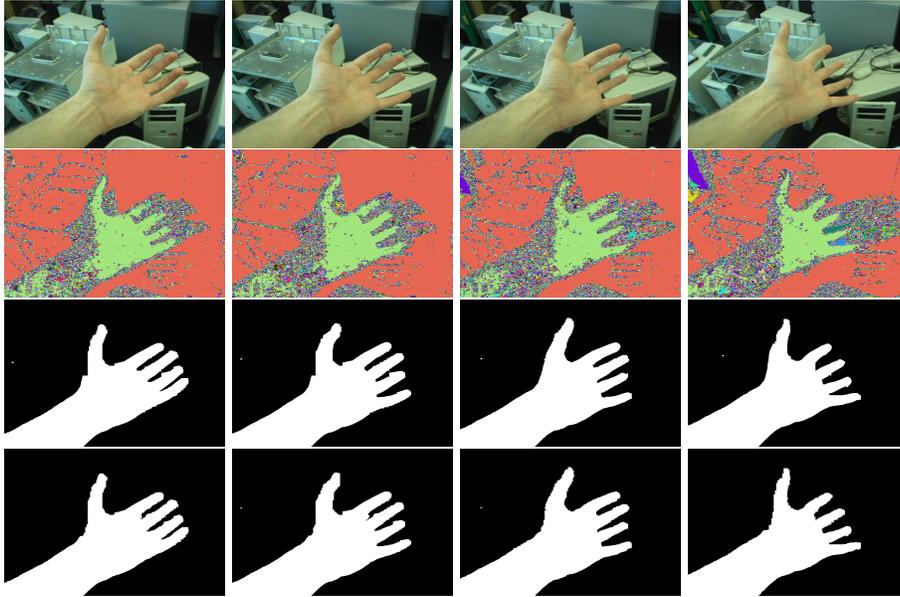


Fig. 4: Variable grouping for video segmentation. The columns correspond to the frames 5, 15, 25 and 38 of the hand sequence [27]. Rows: (i) original frame; (ii) variable grouping with the proposed algorithm; (iii) segmentation result solving the full MAP; (iv) segmentation result solving the approximated MAP. The segmentation results are almost identical even if the approximated solution used a Budget of 5%. The ratio of runtime for this example is ≈ 0.1 .

of 5%. Due to the limitations of the BK-algorithm the proposed method greatly extends its applications.

Video-sequences: Our proposed algorithm can also be applied to group variables for the problem of video segmentation. To evaluate the performance of the proposed method we segmented different video-sequences. It can be seen from Figures 4 and 5 that the proposed algorithm achieves a similar segmentation like the full MAP solution with a much smaller budget and a dramatic reduction of runtime. E.g. for the hand video in Figure 4 (200 frames) we reduced the number of variables from 69.1 million to 3.5 million. For comparison of the results the full MAP solution was only computed for 40 frames since solving the full MAP problem for 200 frames was not possible due to memory reasons. The full MAP problem for the KTH-sequence shown in Figure 5 has 7 million variables and the results shown use an budget of approximately 10% resulting in 0.7 million variables with a comparable segmentation result. In all examples we initialized the segmentation with a few strokes in the first frame.

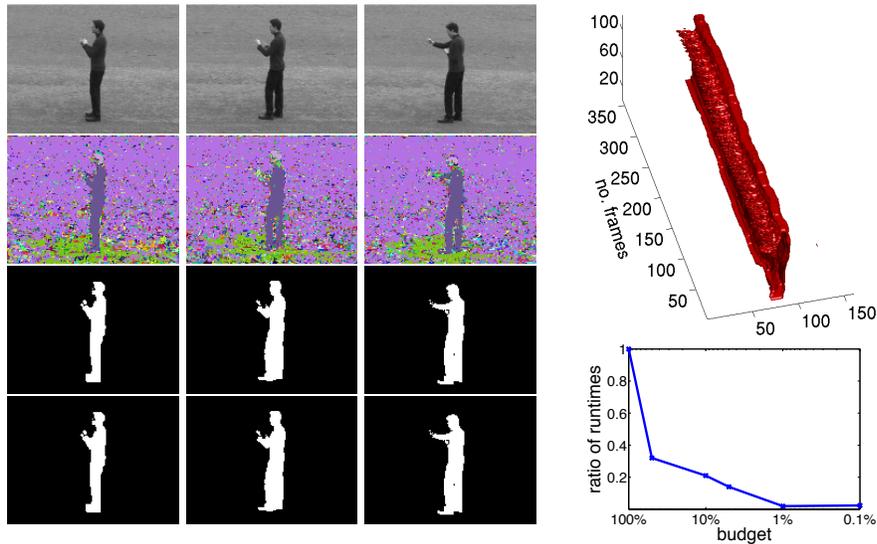


Fig. 5: Variable grouping for video segmentation. The columns correspond to the frames 20, 220 and 350 of the Boxing sequence from [26]. The last column visualizes the isosurface of our segmentation result and the ratio of runtime for a given budget. Rows: (i) original frame; (ii) variable grouping with the proposed algorithm; (iii) segmentation result solving the full MAP; (iv) segmentation result solving the approximated MAP. The segmentation results are almost identical even if the approximated solution used a Budget of 10%. The ratio of runtime for this example is ≈ 0.21 .

5 Conclusion

We presented an efficient algorithm for graph simplification of maximum a posteriori problems that is widely applicable to MAP inference problems in computer vision. It uses Dempster’s theory of evidence and new constraints for the graph based grouping to group large homogeneous regions to one single variable of the problem. In our experiments on segmentation we demonstrated that the segmentation error using the proposed method is smaller or comparable to the full MAP solution. In several experiments on large-scale problems with millions of variables we demonstrate that the reduction in runtime is dramatic while the segmentation quality stays comparable.

References

1. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *TPAMI* **26** (2004) 1124–1137
2. Blake, A., Rother, C., Brown, M., Perez, P., Torr, P.: Interactive image segmentation using an adaptive GMMRF model. In: *ECCV*. (2004) 428–441

3. Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. *ICCV* (2003) 26–33
4. Lempitsky, V., Boykov, Y.: Global optimization for shape fitting. In: *CVPR*. (2007) 1–8
5. Kohli, P., Torr, P.H.S.: Efficiently solving dynamic markov random fields using graph cuts. In: *ICCV*. Volume 2. (2005) 922–929
6. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: interactive foreground extraction using iterated graph cuts. *SIGGRAPH* **23** (2004) 309–314
7. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. *TPAMI* **23** (2002) 1222–1239
8. Delong, A., Boykov, Y.: A scalable graph-cut algorithm for N-D grids. In: *CVPR*. (2008)
9. Kim, T., Nowozin, S., Kohli, P., Yoo, C.D.: Variable grouping for energy minimization. In: *CVPR*. (2011) 1913–1920
10. Bhusnurmath, A., Taylor, C.: Graph cuts via l_1 norm minimization. *TPAMI* **30** (2008) 1866–1871
11. Komodakis, N.: Towards more efficient and effective LP-based algorithms for MRF optimization. In: *ECCV*. (2010) 520–534
12. Boykov, Y., Jolly, M.: Interactive graph cuts for optimal boundary & region segmentation of objects in ND images. In: *ICCV*. Volume 1. (2001) 105–112
13. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. *IJCV* **59** (2004) 167–181
14. Comaniciu, D., Meer, P., Member, S.: Mean shift: a robust approach toward feature space analysis. *TPAMI* **24** (2002) 603–619
15. Levinstein, A., Stere, A., Kutulakos, K.N., Fleet, D.J., Dickinson, S.J., Siddiqi, K.: TurboPixels: fast superpixels using geometric flows. *TPAMI* **31** (2009) 2290–2297
16. Veksler, O., Boykov, Y., Mehrani, P.: Superpixels and supervoxels in an energy optimization framework. In: *ECCV*. (2010) 211–224
17. Scheuermann, B., Rosenhahn, B.: SlimCuts: graphcuts for high resolution images using graph reduction. In: *EMMCVPR*. (2011) 219–232
18. Lermé, N., Létocart, L., Malgouyres, F.: Reduced graphs for min-cut/max-flow approaches in image segmentation. *ENDM* **37** (2011) 63–68
19. Puzicha, J., Buhmann, J.: Multiscale annealing for grouping and unsupervised texture segmentation. *IJCVIU* **76** (1999) 213–230
20. Kohli, P., Lempitsky, V., Rother, C.: Uncertainty driven multi-scale optimization. *DAGM* (2010) 242–251
21. Sinop, A., Grady, L.: Accurate banded graph cut segmentation of thin structures using laplacian pyramids. *MICCAI* **4191** (2006) 896–903
22. Dempster, A.P.: A generalization of Bayesian inference. *Journal of the Royal Statistical Society. Series B (Methodological)* **30** (1968) 205–247
23. Shafer, G.: A mathematical theory of evidence. Princeton university press (1976)
24. Adamek, T., O’Connor, N.E.: Using Dempster-Shafer theory to fuse multiple information sources in region-based segmentation. In: *ICIP*. (2007) 269 – 272
25. Chaabane, S.B., Sayadi, M., Fnaiech, F., Brassart, E.: Dempster-Shafer evidence theory for image segmentation: application in cells images. *IJSP* (2009)
26. Schüldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. In: *ICPR*. (2004) 32–36
27. Sand, P., Teller, S.J.: Particle video: long-range motion estimation using point trajectories. In: *CVPR*. (2006) 2195–2202