Joint optimization of segmentation and appearance models

Sara Vicente Vladimir Kolmogorov
University College London, UK
{S.Vicente,V.Kolmogorov}@cs.ucl.ac.uk

Carsten Rother
Microsoft Research, Cambridge, UK
carrot@microsoft.com

Abstract

Many interactive image segmentation approaches use an objective function which includes appearance models as an unknown variable. Since the resulting optimization problem is NP-hard the segmentation and appearance are typically optimized separately, in an EM-style fashion. One contribution of this paper is to express the objective function purely in terms of the unknown segmentation, using higher-order cliques. This formulation reveals an interesting bias of the model towards balanced segmentations. Furthermore, it enables us to develop a new dual decomposition optimization procedure, which provides additionally a lower bound. Hence, we are able to improve on existing optimizers, and verify that for a considerable number of real world examples we even achieve global optimality. This is important since we are able, for the first time, to analyze the deficiencies of the model. Another contribution is to establish a property of a particular dual decomposition approach which involves convex functions depending on foreground area. As a consequence, we show that the optimal decomposition for our problem can be computed efficiently via a parametric maxflow algorithm.

1. Introduction

The problem of interactive image segmentation has been widely studied and is part of any commercial image editing package. The task is to separate an image into two distinct regions, foreground and background, with additional user guidance. The ultimate goal is to minimize the user interaction and to maximize the quality of the segmentation. Therefore, it is very common to exploit appearance models, e.g. [5, 15, 11, 14], to distinguish better the foreground from the background segment. An appearance model is a statistical model for the color, texture, etc. of the pixels of the segment. The usage of appearance models can be broadly categorized into three classes:

- The appearance is known *apriori*. An example are medical images where the pixels intensity of the foreground often follow a known distribution. Another scenario is semantic segmentation, e.g. an object recognition and segmentation system.
- The appearance is *fixed*, based on the user input. For instance [5, 14] used color model from user-labeled pixels.
- The appearance is *optimized jointly* with the segmentation [15, 10, 1], allowing to choose from a set of models the one that best fits the segments.

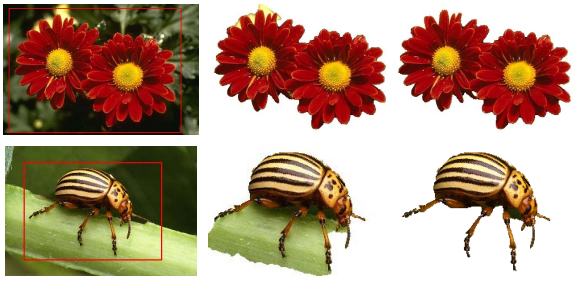
This paper considers the most challenging third option. The advantage is that in the absence of apriori information (first option) it requires in general less user input than option two. For example, it was demonstrated in [15] that good segmentations can be obtained with just an input rectangle containing the object. Note, the second option cannot be used in this case since there is no foreground training data. Furthermore, for the case of "sparse" user input, e.g. a few fore- and background brush strokes, it is very likely that optimizing appearance models is an advantage, as [15] suggests.[1]

The subject of this paper is to study the problem of jointly optimizing segmentation and appearance. For simplicity we consider color as the only appearance feature, which is however, not a limitation of our approach. As input we use a simple rectangle containing the object, as in [15], and show that we outperform on average their EM-style optimization approach. Fig. 1 gives two examples.

The paper is structured as follows. Sec. 2 introduces the problem formulation and discusses further related work. In sec. 3 the problem is rewritten using an energy with higher-order cliques, in a way that the segmentation is the only unknown parameter. The new formulation reveals an interesting, previously unknown, bias of the model towards balanced segmentations, i.e. the preference of fore- and background segments to have similar area. In that section we also show that two existing approaches [1, 12] can be reformulated and optimized in the same way as our model. Then in sec. 4 we discuss how to optimize our energy. The method presented relies on the parametric maxflow algorithm and can escape from the local minima of an EM-style algorithm. It also provides a lower bound and we show that

[1] We confirmed this experimentally by using the publicly available GrabCut dataset [15] which also provides a trimap.

in practice the bound is often tight. For cases where we do not reach global optimality, we introduce a semi-global iterative method in sec. 5 which in practice nearly always improves on the EM technique. The experimental sec. 6 investigates our approach on a large dataset. For cases where we reach global optimality, we will draw conclusions about limitations of the model.



a) User input b) EM solution c) Our solution

Figure 1. *Given the image and bounding box in a) the Expectation Maximization method produces result in b). Our new algorithm gives the segmentation in c) which is not only visually better but also has a lower energy (for the first image it is the global optimum of our energy formulation).*

2. Problem Formulation

We will follow the standard MAP-MRF approach for image segmentation [5, 4, 15, 10]. Hence, we use an energy function of the form

$$E(\boldsymbol{x},\theta^0,\theta^1) = \underbrace{\sum_{p\in V} -\log \Pr(z_p|\theta^{x_p})}_{U(\boldsymbol{x},\theta^0,\theta^1)} + \underbrace{\sum_{(p,q)\in N} w_{pq}|x_p - x_q|}_{P(\boldsymbol{x})} . \tag{1}$$

Here V is the set of pixels, N is the set of neighboring pixels, and $x_p \in \{0,1\}$ is the segmentation label of pixel p (where 0 corresponds to background and 1 to foreground). The first term of equation (1) is the likelihood term, where z_p is the RGB color at site p and θ^0 and θ^1 are respectively the background and foreground color models. The second term is the contrast sensitive edge term, which we define explicitly in sec. 6.

Color modeling Many different models were suggested in the literature. Two popular ones are histograms [5] and Gaussian Mixture Models (GMMs) [4, 15]. We will use the former model, since the use of histograms will be essential for our approach. (Note, it is well-known that MAP estimation with the GMM model is strictly speaking an ill-posed problem since by fitting a Gaussian to the color of a single pixel we may get an infinite likelihood - see [3], section 9.2.1.) As mentioned above, we express the color models in the form of histograms. We assume that the histogram has K bins indexed by $k = 1,\ldots,K$. The bin in which pixel p falls is denoted as b_p, and $V_k \subseteq V$ denotes the set of pixels assigned to bin k. The vectors θ^0 and θ^1 in $[0,1]^K$ represent the distribution over fore- and background, respectively, and sum to 1. The likelihood model is then given by

$$U(\boldsymbol{x},\theta^0,\theta^1) = \sum_p -\log \theta^{x_p}_{b_p} . \tag{2}$$

Optimization The main goal of this paper is to study the problem of minimizing energy function (1). As we show in [18], it is NP-hard. An EM-style algorithm was proposed in [15]. It works by iterating the following steps: (i) Fix color models θ^0, θ^1, minimize energy (1) over segmentation $\boldsymbol{x}$. (ii) Fix segmentation $\boldsymbol{x}$, minimize energy (1) over color models θ^0, θ^1. The first step is solved via a maxflow algorithm, and the second one via standard machine learning techniques for fitting a model to data. Each step is guaranteed not to increase the energy, but of course the procedure may get stuck in a local minimum. Two examples are shown in fig. 1.

In order to avoid local minima, a branch-and-bound framework was proposed in [10]. They demonstrated that a global minimum can be obtained for 8 bins, when allowed models (θ^0,θ^1) are restricted to a set with 2^{16} elements. Unfortunately, in general branch-and-bound techniques suffer from an exponential explosion, so increasing the number of bins or using a finer discretization would present a problem for the method in [10]. We are not aware of any existing technique which could assess the performance of the EM when the number of bins is large, or when the space of models (θ^0,θ^1) is unrestricted.

We now present our approach. First, we will rewrite energy (1) so that it solely depends on the unknown segmentation $\boldsymbol{x}$.

3. Rewriting the energy via high-order cliques

Let us denote n^s_k to be the number of pixels p that fall into bin k and have label s, i.e. $n^s_k = \sum_{p\in V_k} \delta(x_p - s)$. All these pixels contribute the same cost $-\log\theta^s_k$ to the term $U(\boldsymbol{x},\theta^0,\theta^1)$, therefore we can rewrite it as

$$U(\boldsymbol{x},\theta^0,\theta^1) = \sum_s \sum_k -n^s_k \log\theta^s_k . \tag{3}$$

It is well-known that for a given segmentation $\boldsymbol{x}$ distributions θ^0 and θ^1 that minimize $U(\boldsymbol{x},\theta^0,\theta^1)$ are simply the empirical histograms computed over appropriate segments:

$$\theta^s_k = \frac{n^s_k}{n^s} \tag{4}$$

where n^s is the number of pixels with label s: $n^s = \sum_{p\in V}\delta(x_p - s)$. Plugging optimal θ^0 and θ^1 into the energy gives the following expression:

$$E(\boldsymbol{x}) = \min_{\theta^0,\theta^1} E(\boldsymbol{x},\theta^0,\theta^1) = \sum_k h_k(n^1_k) + h(n^1) + P(\boldsymbol{x}) \tag{5}$$

$$h_k(n^1_k) = -g(n^1_k) - g(n_k - n^1_k) \tag{6}$$

$$h(n^1) = g(n^1) + g(n - n^1) \tag{7}$$

where $g(z) = z \log z$, $n_k = |V_k|$ is the number of pixels in bin k and $n = |V|$ is the total number of pixels.

It is easy to see that functions $h_k(\cdot)$ are concave and symmetric about $n_k/2$, and function $h(\cdot)$ is convex and symmetric about $n/2$. The form of equation (5) allows an intuitive interpretation of this model. The first term (sum of concave functions) has a preference towards assigning all pixels in V_k to the same segment. The convex part prefers *balanced* segmentations, i.e. segmentations in which the background and the foreground have the same number of pixels.

Bias of the model Let us analyze this preference towards balanced segmentations in more detail. This bias is most pronounced in the extreme case when all pixels are assigned to unique bins, so $n_k = 1$ for all bins. Then all concave terms $h_k(n_k^1)$ are constants, so the energy consists of the convex part $h(n^1)$ and pairwise terms. Note, however, that the bias disappears in the other extreme case when all pixels are assigned to the same bin ($K = 1$); then concave and convex terms cancel each other. The lemma below gives some intuition about intermediate cases.

Lemma 3.1. *Let V_k be the set of pixels that fall in bin k. Suppose that pixels in V_k are not involved in pairwise terms of the energy, i.e. for any $(p, q) \in N$ we have $p, q \notin V_k$. Also suppose that energy* (5) *is minimized under user-provided hard constraints that force a certain subset of pixels to the background and another subset to the foreground. Then there exists a global minimizer $\boldsymbol{x}$ in which all unconstrained pixels in V_k are assigned either completely to the background or completely to the foreground.*

A proof of this lemma is given in [18]. Note that $h_k(0) = h_k(n_k)$, so if pixels in V_k are not involved in hard constraints then in the absence of pairwise terms the labeling of V_k will be determined purely by the convex term $h(n^1)$, i.e. the model will choose the label that leads to a more balanced segmentation.

Related models There is a relation between the model discussed in this paper and a special case of the model proposed in [1]. If the only feature considered in their formulation is color, their model reduces to equation (5) with functions $h_k(n_k^1)$ and $h(n^1)$ defined in the following way:

$$h_k(n_k^1) = -n_k^1 \log\left(\frac{n_k - n_k^1}{n_k^1}\right) - (n_k - n_k^1)\log\left(\frac{n_k^1}{n_k - n_k^1}\right)$$

$$h(n^1) = n^1 \log\left(\frac{n - n^1}{n^1}\right) + (n - n^1)\log\left(\frac{n^1}{n - n^1}\right)$$

In [12] the authors propose to use a measure of distance between histograms to detect a salient object in an image. This approach can also be converted to a sum of concave functions over the histogram bins (the same as equation (5) without the convex part), where $h_k(n_k^1)$ is defined as follows:

$$h_k(n_k^1) = -\frac{(2n_k^1 - n_k)^2}{2n_k}$$

Both these models can be optimized using the dual decomposition approach, described in the following section.

4. Optimization via dual decomposition

The full energy derived in the previous section has the following form:

$$E(\boldsymbol{x}) = \underbrace{\sum_k h_k(n_k^1) + \sum_{(p,q)\in N} w_{pq}|x_p - x_q|}_{E^1(\boldsymbol{x})} + \underbrace{h(n^1)}_{E^2(\boldsymbol{x})} \quad (8)$$

where $h_k(\cdot)$ are concave functions and $h(\cdot)$ is a convex function. Recall that n_k^1 and n^1 are functions of the segmentation: $n_k^1 = \sum_{p\in V_k} x_p$, $n^1 = \sum_{p\in V} x_p$. It can be seen that the energy function is composed of a submodular part ($E^1(\boldsymbol{x})$) and a supermodular ($E^2(\boldsymbol{x})$) part.

As we showed, minimizing function (8) is an NP-hard problem. We will use a *dual decomposition* (DD) technique, which is a popular approach for solving difficult optimization problems [2]. Its idea is to decompose the energy into several subproblems, each of which can be optimized efficiently. Combining the minima of different subproblems gives a lower bound on the original function. The DD approach then tries to maximize this lower bound over different decompositions.

We now apply this technique to our problem. Let us rewrite the energy as

$$E(\boldsymbol{x}) = [E^1(\boldsymbol{x}) - \langle \boldsymbol{y}, \boldsymbol{x}\rangle] + [E^2(\boldsymbol{x}) + \langle \boldsymbol{y}, \boldsymbol{x}\rangle] \quad (9)$$

where $\boldsymbol{y}$ is a vector in $\mathbb{R}^n$, $n = |V|$ and $\langle \boldsymbol{y}, \boldsymbol{x}\rangle$ denotes the dot product between two vectors. In other words, we added unary terms to one subproblem and subtracted them from the other one. This is a standard use of the DD approach for MRF optimization [19, 16, 17, 9, 20].

Taking the minimum of each term in (9) over $\boldsymbol{x}$ gives a lower bound on $E(\boldsymbol{x})$:

$$\begin{aligned}\Phi(\boldsymbol{y}) &= \underbrace{\min_{\boldsymbol{x}}[E^1(\boldsymbol{x}) - \langle \boldsymbol{y}, \boldsymbol{x}\rangle]}_{\Phi^1(\boldsymbol{y})} + \underbrace{\min_{\boldsymbol{x}}[E^2(\boldsymbol{x}) + \langle \boldsymbol{y}, \boldsymbol{x}\rangle]}_{\Phi^2(\boldsymbol{y})} \\ &\leq \min_{\boldsymbol{x}} E(\boldsymbol{x}) \quad (10)\end{aligned}$$

Note that both minima can be computed efficiently. In particular, the first term can be optimized via a reduction to an min s-t cut problem [13]. In section 4.1 we review this reduction and propose one extension.

To get the tightest possible bound, we need to maximize $\Phi(\boldsymbol{y})$ over $\boldsymbol{y}$. Function $\Phi(\cdot)$ is concave, therefore one could use some standard concave maximization technique, such as a subgradient method [2, 16, 17, 9] which is guaranteed to converge to an optimal bound. Note, such decomposition was used as an example in [20] for enforcing the area constraint; the bound was optimized via a max-sum diffusion algorithm.

We will show that in our case the tightest bound can be computed in polynomial time using a parametric maxflow technique [7].

Theorem 4.1. *Suppose that continuous functions $\Phi^1, \Phi^2 : \mathbb{R}^{|V|} \rightarrow \mathbb{R}$ have the following properties:*

(a) $$\Phi^1(\boldsymbol{y} + \delta \cdot \chi_p) \geq \Phi^1(\boldsymbol{y}) + \min_{x \in \{0,1\}} \{-x\delta\} \qquad (11)$$

for all vectors $\boldsymbol{y}$ and nodes $p \in V$, where χ_p is the vector of size $|V|$ with $(\chi_p)_p = 1$ and all other components equal to zero;

(b) $$\Phi^2(\boldsymbol{y}) = \min_{\boldsymbol{x} \in \{0,1\}^{|V|}} E^2(\boldsymbol{x}) + \langle \boldsymbol{y}, \boldsymbol{x} \rangle \qquad (12)$$

where $E^2(\boldsymbol{x}) = h(\sum_{p \in V} x_p)$ and function $h(\cdot)$ is convex on $[0, n]$ where $n = |V|$, i.e. $2h(k) \leq h(k-1) + h(k+1)$ for $k = 1, \ldots, n-1$.

Under these conditions function $\Phi(\boldsymbol{y}) = \Phi^1(\boldsymbol{y}) + \Phi^2(\boldsymbol{y})$ has maximizer $\boldsymbol{y}$ such that $y_p = y_q$ for any $p, q \in V$.

A proof is given in the Appendix. Theorem 4.1 implies that it suffices to consider vectors $\boldsymbol{y}$ of the form $\boldsymbol{y} = \lambda \mathbf{1}$, where $\mathbf{1}$ is the vector in $\mathbb{R}^n$ with components 1. It is easy to see that we can evaluate $\Phi(\lambda \mathbf{1})$ efficiently for **all** values of λ. Indeed, we need to minimize functions

$$E^1(\boldsymbol{x}) - \lambda \langle \mathbf{1}, \boldsymbol{x} \rangle \ , \quad E^2(\boldsymbol{x}) + \lambda \langle \mathbf{1}, \boldsymbol{x} \rangle \ .$$

For the first function we need to solve a *parametric maxflow* problem [7]. The result is a nested sequence of m solutions $\boldsymbol{x}$, $2 \leq m \leq n+1$ and the corresponding $m-1$ breakpoints of λ. Accordingly, function $\min_{\boldsymbol{x}} [E^1(\boldsymbol{x}) - \lambda \langle \mathbf{1}, \boldsymbol{x} \rangle]$ is a piecewise-linear concave function. All solutions and breakpoints can be computed efficiently by a divide-and-conquer algorithm (see e.g. [8] for a review). The second function can also be handled efficiently. It is not difficult to show that if $E^2(\boldsymbol{x}) = h(\sum_{p \in V} x_p)$ then $\min_{\boldsymbol{x}} [E^2(\boldsymbol{x}) + \lambda \langle \mathbf{1}, \boldsymbol{x} \rangle]$ is a piecewise-linear concave function with breakpoints $h(n-1) - h(n)$, $h(n-2) - h(n-1)$, ..., $h(0) - h(1)$. This implies that $\Phi(\cdot)$ is a piecewise-linear concave function with at most $2n$ breakpoints. In our implementation we construct this function explicitly; after that computing the tightest lower bound (i.e. the maximum of $\Phi(\cdot)$) becomes trivial. Note, however, that this is not the most efficient scheme: in general, maximizing a concave function does not require evaluating **all** breakpoints.

It remains to specify how to get labeling $\boldsymbol{x}$. From the sequence of solutions obtained using parametric maxflow, we choose the one with minimum energy to be the solution for the original problem.

Extended decompositions The decomposition (9) could potentially be strengthened by using other terms. We started to investigate the following decomposition:

$$\begin{aligned} E(\boldsymbol{x}) &= [E^1(\boldsymbol{x}) - \langle \boldsymbol{y}, \boldsymbol{x} \rangle - \sum_k y_k h_k(n_k^1)] \\ &+ [E^2(\boldsymbol{x}) + \langle \boldsymbol{y}, \boldsymbol{x} \rangle + \sum_k y_k h_k(n_k^1)] \end{aligned} \qquad (13)$$

As before, both terms can be minimized in polynomial time assuming that $y_k \leq 1$ for all k and subsets V_k are disjoint. (Note, the second term can be optimized using dynamic programming[2].) Unfortunately, in preliminary tests on a few examples we did not manage to get tighter bounds compared to decomposition (9). Exploring such decompositions is left as a future work.

4.1. Minimizing submodular functions with concave higher order potentials

The decomposition approach described above requires minimizing functions of the form

$$f(\boldsymbol{x}) = \sum_p f_p(x_p) + \sum_{(p,q)} f_{pq}(x_p, x_q) + \sum_k h_k(n_k^1) \qquad (14)$$

where terms $f_{pq}(\cdot, \cdot)$ are submodular and $h_k(\cdot)$ are concave functions. Since each function $h_k(n_k^1)$ is dependent on the labels of all nodes in V_k, these functions correspond to potentials defined over higher order cliques. It is known that the problem of minimizing $f(\cdot)$ can be be reduced to a min s-t cut problem [13]. Let us review how this reduction works. Consider term h_k defined over subset V_k. $h_k(\cdot)$ needs to be defined only for values $0, 1, \ldots, n_k = |V_k|$ so we can assume without loss of generality that $h_k(\cdot)$ is a piecewise-linear concave function with β_k integer breakpoints. The method in [13] first represents the function as a sum of β_k piecewise-linear concave functions with *one* breakpoint. For each function we add an auxiliary variable which is connected to the source or to the sink and to all nodes in V_k. Thus, the number of added edges is $O(n_k \beta_k)$.

In our case function $h_k(\cdot)$ is strictly concave, which implies $\beta_k = O(n_k)$. Thus, the method would add $O((n_k)^2)$ edges. This makes it infeasible in practice for large n_k; even keeping edges in memory would be a problem.

We used the following iterative technique instead. Let us approximate $h_k(\cdot)$ with a piecewise-linear concave function $\bar{h}(\cdot)$ whose set of breakpoints $\mathcal{B}_k$ satisfies $\{0, n_k\} \subseteq \mathcal{B}_k \subseteq \{0, 1, \ldots, n_k\}$. We require $\bar{h}_k(B) = h_k(B)$ for every breakpoint $B \in \mathcal{B}_k$. Using this property, we can uniquely reconstruct function $\bar{h}(\cdot)$ from the set $\mathcal{B}_k$. It is not difficult to see that $\bar{h}(B) \leq h(B)$ for all integer values of B in $[0, n_k]$.

We initialize sets $\mathcal{B}_k$ with a small number of breakpoints, namely $\{0, \lfloor n_k/2 \rfloor, n_k\}$. We then iterate the following procedure: (1) minimize function (14) in which terms $h_k(n_k^1)$ are replaced with approximations $\bar{h}_k(n_k^1)$; obtain optimal solution $\boldsymbol{x}$ and corresponding counts n_k^1; (2) for each bin k set $\mathcal{B}_k := \mathcal{B}_k \cup \{n_k^1\}$. We terminate if none of the sets $\mathcal{B}_k$ change in a given iteration.

[2] The algorithm computes recursively the following quantities for subsets $\mathcal{K} \subseteq \{1, \ldots, K\}$ and integers $m \in [0, \sum_{k \in \mathcal{K}} |V_k|]$:

$$C_{\mathcal{K}}(m) = \min_{\boldsymbol{x}: \sum_k n_k^1 = m} \sum_{k \in \mathcal{K}} \left[\sum_{p \in V_k} y_p x_p + y_k h_k(n_k^1) \right]$$

If $\mathcal{K} = \{k\}$ then $C_{\mathcal{K}}(m)$ can be computed by sorting values y_p, $p \in V_k$. Array $C_{\mathcal{K}' \cup \mathcal{K}''}$ for disjoint subsets $\mathcal{K}'$, $\mathcal{K}''$ can be obtained via *min-convolution* of arrays $C_{\mathcal{K}'}$ and $C_{\mathcal{K}''}$.

This technique must terminate since sets $\mathcal{B}_k$ cannot grow indefinitely. Let $\boldsymbol{x}$ be the labeling produced by the last iteration. It is easy to verify that for any labeling $\boldsymbol{x}'$ there holds

$$f(\boldsymbol{x}') \geq \bar{f}(\boldsymbol{x}') \geq \bar{f}(\boldsymbol{x}) = f(\boldsymbol{x})$$

where $\bar{f}(\cdot)$ is the function minimized in the last iteration. Thus, $\boldsymbol{x}$ is a global minimum of function (14).

5. Semi-global iterative optimization

In our experiments, we observed that for some images the dual decomposition technique performed rather poorly: the number of breakpoints obtained using parametric maxflow was small and none of those breakpoints corresponded to a good solution. In such cases we would probably need to resort to an EM-style iterative technique. In this section we describe how we can use the dual decomposition approach for such iterative minimization.

Suppose that we have a current solution $\bar{\boldsymbol{x}}$. The EM approach would compute empirical histograms $(\bar{\theta}^0, \bar{\theta}^1)$ over $\bar{\boldsymbol{x}}$ using formulas (4) and then minimize energy $E^{\text{EM}}(\boldsymbol{x}) = E(\boldsymbol{x}, \bar{\theta}^0, \bar{\theta}^1)$. We now generalize this procedure as follows. Consider the energy function

$$\bar{E}(\boldsymbol{x}) = (1-\alpha)E^{\text{EM}}(\boldsymbol{x}) + \alpha E(\boldsymbol{x}) \qquad (15)$$

where α is a fixed parameter in $[0, 1]$ and $E(\boldsymbol{x})$ is defined by (5). Note that $\alpha = 0$ gives the energy used by the EM approach, and $\alpha = 1$ gives the global energy (5).

Lemma 5.1. *Suppose that $\boldsymbol{x}$ is a minimizer of $\bar{E}(\cdot)$ for $\alpha \in (0, 1]$ and $\boldsymbol{x}^{EM}$ is a minimizer of $E^{EM}(\cdot)$. Then $E(\boldsymbol{x}) \leq E(\boldsymbol{x}^{EM})$. Furthermore, if $(\bar{\theta}^0, \bar{\theta}^1)$ is computed from some segmentation $\bar{\boldsymbol{x}}$ then $E(\boldsymbol{x}) \leq E(\boldsymbol{x}^{EM}) \leq E(\bar{\boldsymbol{x}})$.*

Proof. Denote $\beta = 1 - \alpha$. Optimalities of $\boldsymbol{x}$ and $\boldsymbol{x}^{\text{EM}}$ imply

$$\begin{aligned} \beta E^{\text{EM}}(\boldsymbol{x}) + \alpha E(\boldsymbol{x}) &\leq \beta E^{\text{EM}}(\boldsymbol{x}^{\text{EM}}) + \alpha E(\boldsymbol{x}^{\text{EM}}) \\ \beta E^{\text{EM}}(\boldsymbol{x}^{\text{EM}}) &\leq \beta E^{\text{EM}}(\boldsymbol{x}) \end{aligned}$$

Adding these inequalities and canceling terms gives $\alpha E(\boldsymbol{x}) \leq \alpha E(\boldsymbol{x}^{\text{EM}})$, or $E(\boldsymbol{x}) \leq E(\boldsymbol{x}^{\text{EM}})$ since $\alpha > 0$. It is well-known that E and M steps of the EM method do not increase the energy; this implies the second claim $E(\boldsymbol{x}^{\text{EM}}) \leq E(\bar{\boldsymbol{x}})$. □

The lemma suggests a semi-global iterative optimization approach in which the next solution is obtained by minimizing function $\bar{E}(\cdot)$ for some value of α. (Clearly, techniques discussed in section 4 are applicable to function (15) as well). We can expect that for sufficiently small values of α the DD approach will produce a global minimum of $\bar{E}(\cdot)$; this is certainly true for $\alpha = 0$.

6. Experimental results

In this section we present experimental results using the previously described model and optimization procedure for interactive image segmentation. We first give some implementation details. The pairwise potentials for the 8-connected grid are defined similar to [5]: $w_{pq} = \frac{\left(\lambda_1 + \lambda_2 \exp -\beta \|z_p - z_q\|^2\right)}{\text{dist}(p,q)}$ with $\lambda_1 = 1$, $\lambda_2 = 10$ and $\beta = \left(2\left\langle (z_p - z_q)^2 \right\rangle\right)^{-1}$, where $\langle\cdot\rangle$ denotes expectation over the image. The histograms are computed over the RGB color space divided into 16^3 bins of equal size.

We report results for the GrabCut database [15] of 49 images with associated user defined bounding box.[3] The outside of the box is constrained to be background. In order to run many experiments we downscaled each image to a maximum side-length of 250 pixels. Careful inspection showed that this did not affect the quality of the results. The running time for these downscaled images is shown in table 1. Note, we did not put much effort into optimizing the code since efficiency was not the primary focus of this paper[4]. We believe that the running times can be considerably reduced.

	Min	Mean	Max
Runtime (in seconds)	27	576	16125
# breakpoints	17	90.5	144
# breakpoints (30 pixel diff)	3	20	38

Table 1. Number of breakpoints and running time.

Dual decomposition for energy minimization Table 2 compares two different methods for energy minimization: our dual decomposition approach and the EM-style procedure used in previous approaches [15, 1]. For 80% of the images our method obtains an energy which is lower than EM, and achieves global optimality for 61% of the images. The two methods obtain the same solution for 4% of the images.[5]

	Dual decomp.	EM
Lower energy	79.6%	16.3%
Global optimum	61.2%	4.1%
Error	10.5%	8.1%
Error (glob. opt. set)	4.1%	4.7%

Table 2. Comparison between the dual decomposition approach proposed in this paper and EM procedure.

Table 1 shows the average number of breakpoints obtained using dual decompositon and also the number of those breakpoints that differ by at least 30 pixels. The small number of breakpoints obtained for some images affects negatively the performance of the dual decomposition method.

We consider as error rate the number of misclassified pixels over the size of the inference region. The average

[3] We exclude the "cross" image since the bounding box covers the whole image.

[4] We use unoptimized C++/Matlab. Flow is not reused. Also we perform parametric maxflow iterations in the inner loop of the procedure in section 4.1 rather than in the outer loop; the latter could be faster. Finally, the maxflow algorithm that we used [6] does not appear to handle well nodes with high degree.

[5] These were the images where EM achieves the global optimum.

error rate is 10.5% for dual decomposition, while for EM it is 8.1%. The dual decomposition method fails for camouflage images and the average error rate is greatly affected by these examples. Figure 2 shows one of these failure cases. If we consider only those images where the global optimum solution is achieved, the error rate drops to 4.1%. For this set of images EM achieves an error of 4.7%, which shows the advantage of achieving global optimality.

Figure 2. *Failure case for dual decomposition (DD) method. The method performs poorly for camouflage images.*

Motivated by the difficult cases we proposed the semi-global method (sec. 5) that uses dual decomposition in an iterative procedure. To show that this method is more powerful than EM we take solution $\bar{\boldsymbol{x}}$ to be an EM fixed point, i.e. EM procedure can not reduce further the energy. We then run the semi-global method and report how often it improves the energy, i.e. escapes from that local minimum.

Recall that the method uses parametric maxflow which produces a sequence of solutions. From this sequence, the final solution can be chosen in two different ways: the solution with smaller energy with respect to the original energy $E(\boldsymbol{x})$ (solution 1, eq. 5) or the solution that minimizes energy $\bar{E}(\boldsymbol{x})$ (solution 2, eq. 15). In table 3 we report results for this procedure. The first two columns show the number of times that the semi-global method improves over EM, considering, respectively, solution 1 and solution 2. The last column shows the number of times that global optimality is verifiably achieved. Recall that $\alpha = 0$ in (15) gives the energy of the EM and $\alpha = 1$ gives the global energy (5). As expected, energy $\bar{E}(\boldsymbol{x})$ is globally minimized more often for smaller values of α.

In the final experiment we were interested to achieve lowest energy for the whole data set with our procedure. For this we consider those 19 images where we do not achieve global optimality, and choose the solution with better energy between EM and dual decomposition to obtain an initial solution. Then we run sequentially the semi-global method for $\alpha = 0.75, 0.5, 0.25, 0$. Each run was initialized with the lowest energy result from the previous run. We conclude that we improved over EM in terms of energy in all examples, and the error rate on the total set reduced to 7.2%.

Model properties and limitations As discussed above, one of the main benefits of the dual decomposition optimization method is that it provides a lower bound. This allows to verify for which images we get global optimality and consequently to analyze the properties and limitations

	Solution 1	Solution 2	Global optimum
$\alpha = 0.25$	91.8%	61.2%	95.9%
$\alpha = 0.5$	95.9%	73.5%	85.7%
$\alpha = 0.75$	89.8%	85.7%	79.6%

Table 3. Results for the semi global method.

of the model. For this purpose, we consider the restricted set of 30 images where dual decomposition provides the global minimum. Fig. 3 shows some results that are visually good.

For some images the current model is not good enough. There are many ways to improve the model. One possible direction is to incorporate other types of appearance models [1]. For some examples, however, we suspect that an improvement can only be achieved by requesting further user input.

In the following we investigate one potential improvement of the model. In sec. 3 we discussed the bias of the model towards balanced segmentations given by the convex term $h(n^1)$. This means that it prefers that the area of the foreground is half of the image size. This bias is important to overcome a degenerate case when user input is limited. More precisely, if the user input is a bounding box then without the term $h(n^1)$ the solution with all pixels being background has the lowest energy.

Let us extend the model by introducing a weight λ for the convex term:

$$E(\boldsymbol{x}) = \sum_k h_k(n_k^1) + \lambda h(n^1) + P(\boldsymbol{x}) \qquad (16)$$

Note that changing the weight does not change the dual decomposition procedure; instead, it changes the criterion according to which we select the solution from the sequence of solutions produced by parametric maxflow. Thus, the method does not need to be run again.

Figure 4 shows the effect of changing this weight. In this example, the solution with smallest error rate does not correspond to the solution with $\lambda = 1$ (our original model). This suggests that the model can be improved by choosing a better weight. More examples are shown in fig. 5.

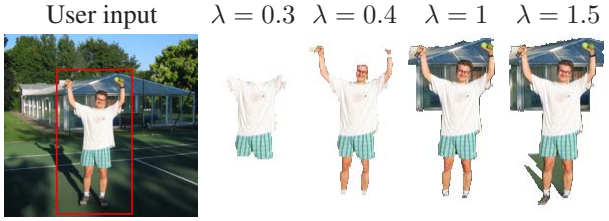


Figure 4. *Choosing a different weight for the bias towards balanced segmentations produces different solutions. All solutions are global optima for the respective weight.*

To show the possible gain of selecting the correct λ we did the following experiment: from the sequence of solutions of parametric maxflow we chose the one with minimum error rate. This solution is guaranteed to be the best

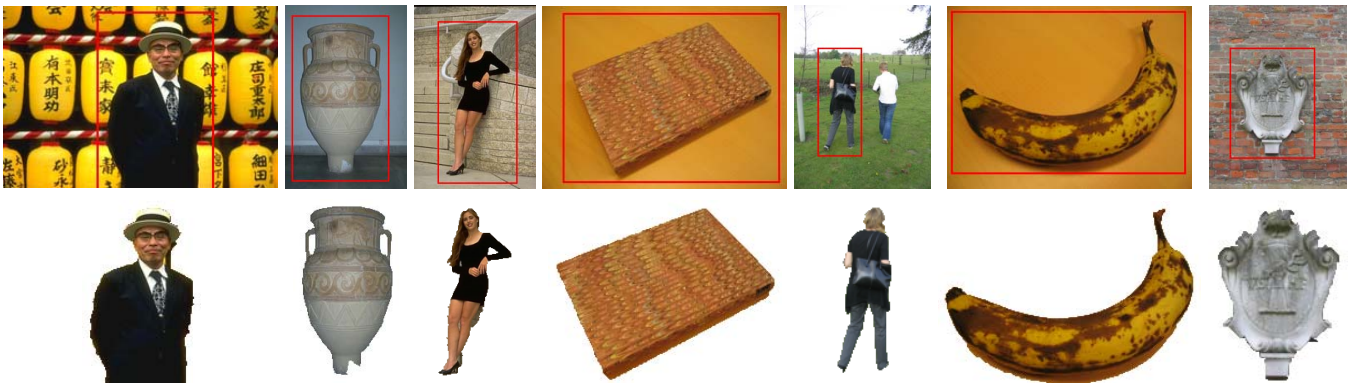


Figure 3. **Global optimum results.** *The first row shows the user input and the second row the segmentation obtained using dual decomposition. For all these images the solution corresponds to the global optimum of the energy.*

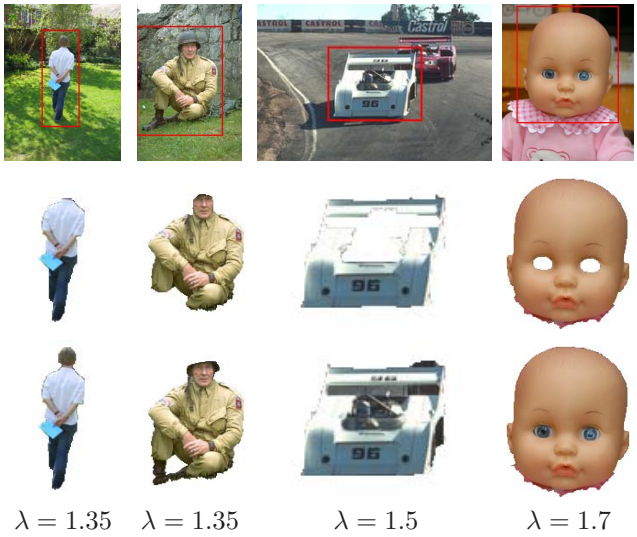


Figure 5. *Model failure cases that can be improved by choosing an appropriate weight λ. The second row corresponds to the solution for the original model with $\lambda = 1$. All solutions are global optima for the respective weight.*

solution obtained using dual decomposition method for a certain fixed λ (but not necessarily the global optimum of eq. 16 for a fixed λ). By selecting this solution the error drops from 4.1% to 3%. In practice the optimal weight could potentially be learned from ground truth segmentations. (Note, the weight would probably depend on the location of the bounding box inside the image). Another option is to have the user selecting the weight, e.g. by browsing through the sequence of solutions.

7. Conclusions and future work

We showed that dual decomposition is a powerful method for optimizing functions of the form of equation (5). It improves over EM-style techniques and it can be combined with this approach for complex images. It also computes the global optimum for 60% of the instances considered. This allows a proper evaluation of this type of model that was not possible before. It reveals that the model can be improved by weighting the bias towards balanced segmentations.

On the optimization side, we showed how to speed up dual decomposition techniques involving convex terms of the area (see theorem 4.1). We hope that this observation will turn out to be useful for other vision applications. Some examples, e.g. a constraint on segmentation area, have been discussed recently in [20, 21].

References

[1] S. Bagon, O. Boiman, and M. Irani. What is a good image segment? a unified approach to segment extraction. In *ECCV (4)*, pages 30–44, 2008.

[2] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[3] C. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, 2008.

[4] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive GMMRF model. In *ECCV*, 2004.

[5] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *ICCV*, 2001.

[6] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 26(9), Sept. 2004.

[7] G. Gallo, M. D. Grigoriadis, and R. E. Tarjan. A fast parametric maximum flow algorithm and applications. *SIAM J. Computing*, 18:30–55, 1989.

[8] V. Kolmogorov, Y. Boykov, and C. Rother. Applications of parametric maxflow in computer vision. In *ICCV*, 2007.

[9] N. Komodakis, N. Paragios, and G. Tziritas. MRF optimization via dual decomposition: Message-passing revisited. In *ICCV*, 2005.

[10] V. S. Lempitsky, A. Blake, and C. Rother. Image segmentation by branch-and-mincut. In *ECCV (4)*, 2008.

[11] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. *SIGGRAPH*, August 2004.

[12] T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. In *CVPR*, 2007.

[13] P.Kohli, L.Ladicky, and P.Torr. Robust higher order potentials for enforcing label consistency. In *CVPR*, 2008.

[14] A. Protiere and G. Sapiro. Interactive image segmentation via adaptive weighted distances. *IEEE Transactions on Image Processing*, 16(4):1046–1057, April 2007.

[15] C. Rother, V. Kolmogorov, and A. Blake. Grabcut - interactive foreground extraction using iterated graph cuts. *SIGGRAPH*, August 2004.

[16] M. I. Schlesinger and V. V. Giginyak. Solution to structural recognition (MAX,+)-problems by their equivalent transformations. Part 1. *Control Systems and Computers*, (1):3–15, 2007.

[17] M. I. Schlesinger and V. V. Giginyak. Solution to structural recognition (MAX,+)-problems by their equivalent transformations. Part 2. *Control Systems and Computers*, (2):3–18, 2007.

[18] S. Vicente, V. Kolmogorov, and C. Rother. Joint optimization of segmentation and appearance models. Technical report, UCL, 2009.

[19] M. Wainwright, T. Jaakkola, and A. Willsky. MAP estimation via agreement on trees: Message-passing and linear-programming approaches. *IEEE Trans. Information Theory*, 51(11):3697–3717, 2005.

[20] T. Werner. High-arity interactions, polyhedral relaxations, and cutting plane algorithm for soft constraint optimisation (MAP-MRF). In *CVPR*, 2008.

[21] O. J. Woodford, C. Rother, and V. Kolmogorov. A global perspective on MAP inference for low-level vision. In *ICCV*, 2009.

Appendix: Proof of theorem 4.1

Let $\boldsymbol{y}^\circ$ be a maximizer of $\Phi(\cdot)$, and let Ω be the set of vectors $\boldsymbol{y}$ such that $\Phi(\boldsymbol{y}) = \Phi(\boldsymbol{y}^0)$ and $\min_{q\in V} y^\circ_q \le y_p \le \max_{q\in V} y^\circ_q$ for $p \in V$. Clearly, Ω is a non-empty compact set. Let $\boldsymbol{y}$ be a vector in Ω with the minimum value of $\Delta(\boldsymbol{y}) = \max_{p\in V} y_p - \min_{p\in V} y_p$. (The minimum is achieved in Ω due to compactness of Ω and continuity of function $\Delta(\cdot)$.) If there are multiple vectors $\boldsymbol{y} \in \Omega$ that minimize $\Delta(\boldsymbol{y})$, we will choose a vector such that the cardinality of the set $\{p \in V : \min_{q\in V} y_q < y_p < \max_{q\in V} y_q\}$ is maximized. We need to prove that $\Delta(\boldsymbol{y}) = 0$. Suppose that $\Delta(\boldsymbol{y}) > 0$. Let $p^- \in V$ and $p^+ \in V$ be nodes with the minimum and maximum values of y_p, respectively, so that $y_{p^+} - y_{p^-} = \Delta(\boldsymbol{y}) > 0$.

Denote $\bar{E}^2(\boldsymbol{x}) = E^2(\boldsymbol{x}) + \langle \boldsymbol{x}, \boldsymbol{y}\rangle$, and let $\mathcal{X}$ be the set of minimizers of $\bar{E}^2(\cdot)$. We claim that there exist labelings $\boldsymbol{x}^-, \boldsymbol{x}^+ \in \mathcal{X}$ such that $x^-_{p^-} = 0$, $x^+_{p^+} = 1$. Indeed, suppose that all labelings $\boldsymbol{x} \in \mathcal{X}$ have $x_{p^-} = 1$, then there exists sufficiently small $\delta \in (0, \Delta(\boldsymbol{y}))$ such that increasing y_{p^-} by δ will not affect the optimality of labelings in $\boldsymbol{x} \in \mathcal{X}$. As a result of this update, $\Phi^2(\boldsymbol{y}) = \min_{\boldsymbol{x}}[E^2(\boldsymbol{x}) + \langle \boldsymbol{y}, \boldsymbol{x}\rangle]$ will increase by δ and $\Phi^1(\boldsymbol{y})$ will decrease by no more than δ due to (11), therefore vector $\boldsymbol{y}$ will remain a maximizer of $\Phi(\cdot)$. After this update either $\Delta(\boldsymbol{y})$ will decrease or the cardinality of the set $\{p \in V : \min_{q\in V} y_q < y_p < \max_{q\in V} y_q\}$ will increase. This contradicts to the choice of $\boldsymbol{y}$, which proves the existence of labeling $\boldsymbol{x}^- \in \mathcal{X}$ with $x^-_{p^-} = 0$. Similarly, suppose that all labelings $\boldsymbol{x} \in \mathcal{X}$ have $x_{p^+} = 0$, they will remain optimal if we decrease y_{p^+} by a sufficiently small amount $\delta \in (0, \Delta(\boldsymbol{y}))$. As a result of this update, $\Phi^2(\boldsymbol{y}) = \min_{\boldsymbol{x}}[E^2(\boldsymbol{x}) + \langle \boldsymbol{y}, \boldsymbol{x}\rangle]$ will not change and $\Phi^1(\boldsymbol{y})$ will not decrease, therefore vector $\boldsymbol{y}$ will remain a maximizer of $\Phi(\cdot)$. This contradicts to the choice of $\boldsymbol{y}$, and proves the existence of labeling $\boldsymbol{x}^+ \in \mathcal{X}$ with $x^+_{p^+} = 1$.

Next, we will establish some useful properties about the structure of $\mathcal{X}$. Let us call labeling $\boldsymbol{x} \in \{0,1\}^n$ *monotonic* if it satisfies the following property: if $y_p < y_q$ for nodes $p, q \in V$ then $x_p \ge x_q$. Clearly, any labeling $\boldsymbol{x} \in \mathcal{X}$ must be monotonic. Indeed, if $y_p < y_q$, $x_p = 0$ and $x_q = 1$ then swapping the labels of p and q would decrease $\bar{E}^2(\boldsymbol{x})$.

Let us introduce function

$$\bar{h}(k) = \min_{\boldsymbol{x}:||\boldsymbol{x}||=k} \bar{E}^2(\boldsymbol{x}) = h(k) + \min_{\boldsymbol{x}:||\boldsymbol{x}||=k} \langle \boldsymbol{x}, \boldsymbol{y}\rangle$$

where we denoted $||\boldsymbol{x}|| = \sum_{p\in V} x_p$. It is easy to see that $\boldsymbol{x} \in \mathcal{X}$ if and only if two conditions hold: (i) $\bar{h}(k)$ achieves the minimum at $k = ||\boldsymbol{x}||$; (ii) labeling $\boldsymbol{x}$ is monotonic. (Note, all monotonic labelings $\boldsymbol{x}$ with the same count $||\boldsymbol{x}||$ have the same value of $\langle \boldsymbol{x}, \boldsymbol{y}\rangle$.)

Let $(y^1, \ldots, y^n)$ be the sequence of values y_p, $p \in V$ sorted in the non-decreasing order. In other words, y^k is the k-th smallest element among values y_p, $p \in V$. Clearly, we have

$$\bar{h}(k) = h(k) + \sum_{i=1}^{k} y^i \ , \qquad k = 0, 1, \ldots, n$$

Functions $h(k)$ and $s(k) = \sum_{i=1}^{k} y^i$ are convex, so $\bar{h}(k)$ is convex as well. Therefore, the set of values of k that minimize $\bar{h}(k)$ form an interval $[k^-, k^+]$ where $0 \le k^- \le k^+ \le n$. Furthermore, if $k^- < k^+$ then $y^{k^-+1} = y^{k^+}$. Indeed, we have $\bar{h}(k) = const$ for $k \in [k^-, k^+]$, i.e. function $\bar{h}(\cdot)$ is linear on $[k^-, k^+]$. It is a sum of two convex functions, so both functions must be linear on $[k^-, k^+]$. This implies that $s(k^-+1) - s(k^-) = s(k^+) - s(k^+-1)$, i.e. $y^{k^-+1} = y^{k^+}$.

Let us show that $y_{p^+} = y^{k^+}$. Suppose not: $y^{k^+} < y_{p^+}$. Then there are at least k^+ nodes $p \in V$ with $y_p < y_{p^+}$. They must satisfy $x^+_p = 1$, since $x^+_{p^+} = 1$ and $\boldsymbol{x}^+$ is monotonic. Thus, there are at least $k^+ + 1$ nodes $p \in V$ with $x_p = 1$, so $||\boldsymbol{x}^+|| \ge k^+ + 1$ - a contradiction.

Similarly, we can show that $y_{p^-} = y^{k^-+1}$. (Note, we have $k^- \le ||\boldsymbol{x}^-|| \le n - 1$.) Suppose not: $y_{p^-} < y^{k^-+1}$. Then there are at least $n - k^-$ nodes $p \in V$ with $y_p > y_{p^-}$. They must satisfy $x^-_p = 0$, since $x^-_{p^-} = 0$ and $\boldsymbol{x}^-$ is monotonic. Thus, there are at least $n - k^- + 1$ nodes $p \in V$ with $x_p = 0$, so $||\boldsymbol{x}^-|| \le k^- - 1$ - a contradiction.

The arguments above imply that if $k^- < k^+$ then $y_{p^-} = y^{k^-+1} = y^{k^+} = y_{p^+}$, and if $k^- = k^+$ then $y_{p^-} = y^{k^-+1} \ge y^{k^+} = y_{p^+}$. This contradicts to the assumption $y_{p^-} < y_{p^+}$ made earlier.