Cosegmentation Revisited: Models and Optimization

Sara Vicente[1], Vladimir Kolmogorov[1], and Carsten Rother[2]

[1] University College London
[2] Microsoft Research Cambridge

Abstract. The problem of cosegmentation consists of segmenting the same object (or objects of the same class) in two or more distinct images. Recently a number of different models have been proposed for this problem. However, no comparison of such models and corresponding optimization techniques has been done so far. We analyze three existing models: the L1 norm model of Rother et al. [1], the L2 norm model of Mukherjee et al. [2] and the "reward" model of Hochbaum and Singh [3]. We also study a new model, which is a straightforward extension of the Boykov-Jolly model for single image segmentation [4].

In terms of optimization, we use a Dual Decomposition (DD) technique in addition to optimization methods in [1,2]. Experiments show a significant improvement of DD over published methods. Our main conclusion, however, is that the new model is the best overall because it: (i) has fewest parameters; (ii) is most robust in practice, and (iii) can be optimized well with an efficient EM-style procedure.

1 Introduction

The task of Figure-Ground segmentation is a widely studied problem in computer vision. Given a *single* image there are techniques that attempt to automatically partition the image into multiple objects and background. If the goal is to have a single object segmented, i.e. a binary segmentation, there is the natural ambiguity of which object is the desired one. In this case interactive segmentation techniques must be considered where the user gives additional hints.

There are many interesting application scenarios where *multiple* images are available. This means each image depicts the "same" foreground object in front of potentially arbitrary backgrounds. In contrast to the single image case, the task of segmenting the common object automatically in all images is now well-defined. This task is called "cosegmentation" and was first addressed in [1]. Let us be more precise on the definition of the "same" foreground object. In this paper we use the definition of [1,2,3] where the only constraint is that the distribution of some appearance features of the foreground region in each image have to be similar. The appearance features can encode different information, like color and texture, and various similarity measures can be envisioned. This definition allows for a wide range of applications. One application is to create a visual summary from personal photo collections, by segmenting automatically all instances of the same object, e.g. a person and a dog [5]. Another application is to use the segmentation

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 465–479, 2010.

of the common object to efficiently edit all occurrences of this object in one step, e.g. by changing its contrast [6]. The practical challenge in the case of segmenting the same object is that distributions may not match exactly, due to changes in illumination, in viewpoint or object (self-)occlusion. Our definition of cosegmentation can potentially also be used for segmenting different objects of the same class. An example of an unsupervised object-class recognition and segmentation system is [7], where more features are used other than appearance, e.g. shape. It can be expected that for most object classes, appearance features alone are not strong enough, hence this application is out of the scope of this paper.

Very recently in [8] the authors used a different formulation of the cosegmentation problem. They casted it into a clustering problem with two cluster. They show results for image pairs and for multiple images of objects of the same class.

It is worth mentioning that several recent papers considered a simplified cosegmentation problem where user interaction is available. In [5] the authors segment several images of the same object, assuming one of those images is hand-segmented. They model local appearance and edge profiles from the segmented image in order to "transduct" such segmentation into the remaining images. In [9,6] the user input is in the form of foreground/background scribbles in one or many images from the collection. In [9] the authors discuss how the choice of the seed image influences the performance of their method. In [6] a way of guiding the user interactions is presented. We envision that the insights of this paper will also help to improve the task of interactive cosegmentation.

The goal of this paper is to examine theoretically and practically different models and optimization methods for cosegmentation. To achieve this we limit ourselves to the task of cosegmenting two images only, with color as the only appearance feature, and where distributions are expressed in terms of histograms. We consider three existing models [1,2,3], which differ only in the distance measure between the two color histograms. We also consider a new model, which is a straightforward extension from a single to multiple images of Boykov-Jolly [4]. For a fair comparison we improved on existing optimization methods for the models in [1,2]. We achieved this by using a *Dual Decomposition* technique. For a quantitative comparison we built a dataset of 100 image-pairs with varying levels of complexity by simulating changes in scale and illumination.

The paper is organized as follows. Section 2 introduces the four different models and discusses some of their properties. Since the optimization for some models is NP-hard, it is important to choose the best possible optimization procedure. In Sect. 3 we review such methods. In Sect. 4 we compare experimentally both the models and the optimization methods and conclude which are the better performing methods.

2 Models

We start this section by introducing some notation:

- $x_p \in \{0,1\}$ is the label for pixel p, where $p \in \mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2$ and $\mathcal{P}_1$, $\mathcal{P}_2$ are respectively the set of pixels in image 1 and image 2. We use letter $k \in \{1,2\}$ for denoting the image number.

- z_p is the appearance of pixel p (e.g. color or texture) and such measurement is quantized into a finite number of bins. Variable b ranges over histogram bins ($b \in \{1, ..., B\}$ where B is the total number of bins), and $\mathcal{P}_{kb}$ denotes the set of pixels p in image k whose measurement z_p falls in bin b.
- h_k is the empirical un-normalized histogram of foreground pixels for image k: it is a vector of size B with components $h_{kb} = \sum_{p \in \mathcal{P}_{kb}} x_p$.

As stated earlier, one of the goals of this paper is to compare different cosegmentation models that have been previously proposed. Such models fit into a single framework, where the cosegmentation problem is formulated as an energy optimization, with an energy of the following form:

$$E(\boldsymbol{x}) = \sum_p w_p x_p + \sum_{(p,q)} w_{pq} |x_p - x_q| + \lambda E^{global}(h_1, h_2) \tag{1}$$

Jointly, the first two terms form the traditional MRF term for both images, where w_p is the unary weight for each pixel and w_{pq} is the pairwise weight. The last term, E^{global}, encodes a similarity measure between the foreground histograms of both images and λ is the weight for that term.

Following [1], we will use a ballooning term for the first term, constant for every pixel: $w_p = \mu$. This biases the solution to one of the possible labels and it is important to prevent trivial solutions (i.e. both images being labeled totally background or foreground). If the bias is not present (i.e. if $w_p = 0$ and the energy does not have unary terms) such trivial solutions are always a global optimum of the energy. Alternatively, in [2,3] the authors used user interaction to compute pixel-dependent unary terms [10]. We are interested in automatic cosegmentation so unary terms based on user interaction are not available.

The second term is a contrast sensitive smoothness term whose weight is given by $w_{pq} = \frac{\left(\lambda_i + \lambda_c \exp - \beta \|z_p - z_q\|^2\right)}{\mathrm{dist}(p,q)}$ with $\beta = \left(2\left\langle (z_p - z_q)^2 \right\rangle\right)^{-1}$, where $\langle \cdot \rangle$ denotes expectation over the image and λ_i, λ_c are respectively the weight for Ising prior and for the contrast sensitive term.

The models differ in the way the term E^{global} in equation (1) is defined.

Model A: L1-norm. This model was first introduced in [1] and it was derived from a generative model. The global term in the energy was defined as follows:

$$E^{global} = \sum_b |h_{1b} - h_{2b}| \tag{2}$$

where the L1-norm is used to compute foreground histograms similarity.

Model B: L2-norm. This formulation was introduced in [2] and it was defined as follows:

$$E^{global} = \sum_b (h_{1b} - h_{2b})^2 \tag{3}$$

It is similar to the previous formulation in equation (2), with the difference that the norm used to measure histogram similarity is the L2-norm instead of the L1-norm. The authors motivate this change by arguing that such a model has some interesting properties and allows the use of alternative optimization methods.

Model C: Reward model. In [3] the authors used the following global term:

$$E^{global} = -\sum_b h_{1b} \cdot h_{2b} \tag{4}$$

They motivate the use of such a model by replacing the penalization term with a rewarding term.

Recall that the original formulation in [2,3] uses pixel-dependent unary terms, while we use a constant ballooning force: $w_p = \mu$.

Both model A and model B lead to NP-hard optimization problems [1], while model C leads to a submodular problem that can be efficiently optimized with graph cuts [3].

Model D: Boykov-Jolly model. The last model that we consider is a natural extension of the generative model for binary image segmentation in [4,1,11]. These papers use a separate appearance model for each of the two regions (background and foreground). In our case we have *three* regions - two separate backgrounds and one common foreground. Accordingly, we introduce three appearance models - θ_1^B, θ_2^B and θ^F. This leads to a generative model with the posterior described by the following energy function:

$$E\left(\boldsymbol{x}, \theta_1^B, \theta_2^B, \theta^F\right) = \sum_{(p,q)} w_{pq}|x_p - x_q| + \lambda \sum_k \sum_{p \in \mathcal{P}_k} U(x_p, \theta_k^B, \theta^F) \tag{5}$$

where

$$U(x_p, \theta^B, \theta^F) = \begin{cases} -\log(Pr(z_p|\theta^F)) & \text{if } x_p = 1 \\ -\log(Pr(z_p|\theta^B)) & \text{if } x_p = 0 \end{cases} \tag{6}$$

Since we are interested in automatic cosegmentation, the appearance models θ_1^B, θ_2^B and θ^F are not available in advance. In order to compute them, we minimize energy (5) jointly over segmentation and appearance models using an EM-style technique proposed in [11].

Model D is quite similar to the model used by Batra et al. [6] for interactive cosegmentation; the only difference is that Batra et al. used a single background model for all images. Model D also bears some resemblance to the generative model of Rother et al. [1] but there are some differences. In [1], the motivation was model selection, since two competitive models were considered: one where both images shared the same foreground appearance model and another where they had independent appearance models. The segmentation was then chosen so that the first model had higher posterior probability. In our case, we consider only a single model and try to find jointly the segmentation and appearance models that maximize the posterior probability. This formulation should be more appropriate when we know in advance that the two images have a common object. Also, it appears to lead to a simpler optimization problem: generalizing an EM-style procedure to the model in [1] is not straightforward.

2.1 An Alternative Formulation of Model D

To gain more insights into model D, we express its energy in a different way using the approach in [12]. It is known that for a fixed segmentation $\boldsymbol{x}$, optimal histograms that minimize energy (5) are simply the empirical histograms:

$$\theta_b^F = \frac{h_{1b} + h_{2b}}{H_1 + H_2} \qquad\qquad \theta_{kb}^B = \frac{\overline{h}_{kb}}{\overline{H}_k} \tag{7}$$

where we introduced the following notation: $H_k = \sum_b h_{kb}$ is the total number of foreground pixels in image k, $\overline{h}_{kb} = |\mathcal{P}_{kb}| - h_{kb}$ is the number of background pixels in image k belonging to bin b, and $\overline{H}_k = |\mathcal{P}_k| - H_k$ is the total number of background pixels in image k. Note, all quantities h_{kb}, $\overline{h}_{kb}$, H_k, $\overline{H}_k$ are functions of the segmentation $\boldsymbol{x}$ (recall that $h_{kb} = \sum_{p \in \mathcal{P}_{kb}} x_p$).

Following [12], we plug histograms (7) into the energy (5). Then the energy becomes of the form (1) with no unary terms ($w_p = \mu = 0$) and the following global term:

$$E^{global} = \sum_b \beta\left(h_{1b} + h_{2b}\right) + \sum_{k,b} \beta\left(\overline{h}_{kb}\right) - \beta\left(H_1 + H_2\right) - \sum_k \beta\left(\overline{H}_k\right) \qquad (8)$$

where $\beta(z) = -z \log z$ is a concave function.

In the case of a single image the Boykov-Jolly model prefers assigning pixels in the same bin either entirely to the background or entirely to the foreground [12]; this leads to "compact" histograms. A similar fact holds for model D (the proof is entirely analogous to that in [12]).

Proposition 1. *Function* (8) *has a minimizer* $\boldsymbol{x}$ *such that for each* (k, b)*, pixels in* $\mathcal{P}_{kb}$ *are either all labeled as 0 or all labeled as 1.*

2.2 Remarks on Model Properties

Before presenting an experimental comparison of the models, we would like to give some informal remarks which may give insights into their relative performance. We will first consider models A, B and D, and come back to model C at the end.

We believe that a fundamental difference of model D from other models is that it takes into account the prior knowledge that all regions are represented by compact histograms. For the case of a single image, the bias of the Boykov-Jolly model was discussed in [12]: it prefers segmentations in which pixels that fall in the same bin are assigned to the same segment (background or foreground), and among such segmentations the model picks the most *balanced* one, i.e. the segmentation in which the areas of the background and the foreground match. We conjecture that these properties carry over to the cosegmentation case. It can be shown, for example, that if the two images are identical and all bins are of the same size (i.e. $|\mathcal{P}_{kb}| = const$ for all k, b) then the global term will be minimized by a segmentation in which exactly half of the bins are assigned to the foreground. Due to a bias towards balanced segmentation we did not use the "ballooning force" for model D, i.e. we chose $\mu = 0$, which produced reasonable results. In contrast, the other models required this extra parameter μ in order to avoid trivial solutions.

Unlike model D, models A and B do not impose any penalty if pixels in the same bin, $\mathcal{P}_{kb}$, are assigned to two different segments. We argue that this has both pros and cons, as illustrated by two scenarios below.

Scenario 1. Assume that the background colors do not overlap with the foreground nor with the other background. Furthermore, suppose that the foreground regions in the two images match only partially, for example, due to an illumination change or scaling. Thus, we have $|\mathcal{P}_{kb}| > |\mathcal{P}_{\overline{k}b}|$ for some bin b where $\overline{k} \in \{1,2\}, \overline{k} \neq k$. Models A and B will bias $|\mathcal{P}_{kb}| - |\mathcal{P}_{\overline{k}b}|$ pixels to an incorrect label. In contrast, model D should not be affected; it will assign all pixels in $\mathcal{P}_{kb}$ and $\mathcal{P}_{\overline{k}b}$ to the foreground, as desired.

Scenario 2. Let us now assume that we have "camouflage" in one of the images, i.e. colors of the background and the foreground overlap. Thus, we again have $|\mathcal{P}_{kb}| > |\mathcal{P}_{\overline{k}b}|$, but now the behavior of models A and B will be correct, while model D will try to incorrectly assign *all* pixels in $\mathcal{P}_{kb}$ to the foreground (or to the background).

We conclude that without camouflage model D should cope better with illumination and scale changes than model A and especially than model B. On the other hand, models A and B should be more robust to a camouflage in *one* of the images.

Let us now return to model C. Assume for simplicity that there are no pairwise terms. The energy can then be written as $E(\boldsymbol{x}) = \sum_b E_b(h_{1b}, h_{2b})$ where

$$E_b(h_{1b}, h_{2b}) = \mu(h_{1b} + h_{2b}) - \lambda h_{1b} \cdot h_{2b}$$

We must have $\mu > 0$, otherwise all pixels would be assigned to the foreground. Minimizing E_b over $[0, n_{1b}] \times [0, n_{2b}]$ where $n_{1b} = |\mathcal{P}_{1b}|$, $n_{2b} = |\mathcal{P}_{2b}|$ gives the following rule: if $n_{1b} \cdot n_{2b}/(n_{1b} + n_{2b}) \leq \mu/\lambda$ then assign pixels in $\mathcal{P}_{1b} \cup \mathcal{P}_{2b}$ to the background, otherwise assign these pixels to the foreground. This reliance on the harmonic mean of n_{1b} and n_{2b} can lead to unexpected results (Fig. 1). In our experiments we found that model C performs considerably worse than the other models.

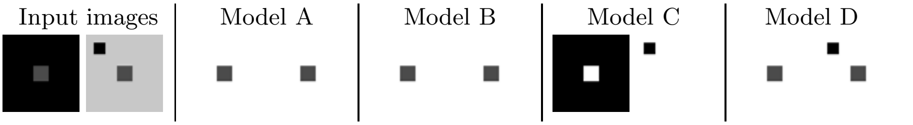

Fig. 1. Synthetic example illustrating the properties of the different models. The input images have only 3 different colors.

3 Optimization Methods

In this section we discuss several optimization methods that can be used for the models discussed in the previous section.

3.1 Trust Region Graph Cut (TRGC)

This method was proposed in [1] for model A and it can be viewed as a discrete analogue of trust region methods for continuous optimization. TRGC can be applied to energy functions of the form $E(\boldsymbol{x}) = E_1(\boldsymbol{x}) + E_2(\boldsymbol{x})$ where $E_1(\boldsymbol{x})$ is submodular and $E_2(\boldsymbol{x})$ is arbitrary. It works by iteratively replacing $E_2(\boldsymbol{x})$

with a linear approximation and it produces a sequence of solutions with the guarantee that in each iteration the energy does not go up.

In [1] the authors used TRGC inside an iterative scheme for cosegmentation that alternated between updating the segmentation for each image individually while the foreground histogram of the other image was fixed. This method requires a segmentation for initialization. In our experiments we observed that its performance is very dependent on that initialization.

We used the implementation of this method from [1]. We also adapted it to model B, i.e. replaced L1 norm with L2 norm.

3.2 Quadratic Pseudo Boolean Optimization

In [2] the authors observed that model B is represented by a **quadratic** pseudo-boolean function. Indeed, histograms h_1 and h_2 depend linearly on $\boldsymbol{x}$: $h_{kb} = \sum_{p \in \mathcal{P}_{kb}} x_p$. Therefore, expanding expression $(h_{1b} - h_{2b})^2$ yields a sum of linear terms and quadratic terms of the form $c_{pq} x_p x_q$, some of which are non-submodular. Mukherjee et al. [2] formulated a linear programming relaxation of the problem, which is equivalent to the roof duality relaxation [13,14] for the quadratic function $E(\boldsymbol{x})$. This relaxation can be solved via a maxflow algorithm, and it yields a partial solution: the nodes are divided into labeled and unlabeled, with the guarantee that the labels of the labeled nodes are optimal. An important question is how to set the segmentation for unlabeled nodes. Mukherjee et al. [2] use the segmentation obtained by minimizing energy $E(\boldsymbol{x})$ without the global term E^{global}. In our experiments we use a constant ballooning force ($w_p = \mu$), so this procedure assigns the same label to all unlabeled nodes.

Note that, model C is also represented by a quadratic function, but unlike the previous case this quadratic function is submodular. Therefore, model C can be optimized exactly by a single call to a maxflow algorithm [3].

3.3 Dual Decomposition (DD)

Dual Decomposition (DD) is a popular technique for solving combinatorial optimization problems [15], which proved to be very successful for MRF optimization [16,17,18,19,12]. The idea of DD is to decompose the original problem into smaller, easier subproblems that can be efficiently optimized. Combining the solution of such subproblems yields a lower bound for the initial problem. This lower bound is then maximized over different decompositions. We applied this technique to models A, B and D as described below.

Dual decompositions for models A and B. Let us write the corresponding optimization problems as follows:

$$\min_{\boldsymbol{x},\boldsymbol{y}} \quad E^{MRF}(\boldsymbol{x}) + \sum_b g(y_b) \tag{9a}$$

$$\text{s.t.} \quad y_b = \sum_{p \in \mathcal{P}_{1b}} x_p - \sum_{p \in \mathcal{P}_{2b}} x_p \equiv \sum_{k,p} a_{bp} x_p \qquad b = 1, ..., B \tag{9b}$$

where g is a convex function: $g(y) = \lambda|y|$ for model A and $g(y) = \lambda y^2$ for model B. Coefficients a_{bp} are defined as follows: $a_{bp} = 1$ if $p \in \mathcal{P}_{1b}$; $a_{bp} = -1$ if $p \in \mathcal{P}_{2b}$ and $a_{bp} = 0$ otherwise.

We form a standard Lagrangian function by relaxing constraints (9b) and introducing a Lagrangian multiplier $\boldsymbol{\theta}$:

$$L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = E^{MRF}(\boldsymbol{x}) + \sum_b g(y_b) + \sum_b \theta_b \left(y_b - \sum_p a_{bp} x_p \right) \tag{10}$$

Minimizing the Lagrangian over $(\boldsymbol{x}, \boldsymbol{y})$ gives a lower bound on the original problem:

$$\Phi(\boldsymbol{\theta}) = \min_{\boldsymbol{x},\boldsymbol{y}} L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) \tag{11a}$$

$$= \min_{\boldsymbol{x}} \left[E^{MRF}(\boldsymbol{x}) - \sum_{p,b} a_{bp} \theta_b x_p \right] + \sum_b \min_{y_b} \left[g(y_b) + \theta_b y_b \right] \tag{11b}$$

$$\Phi(\boldsymbol{\theta}) \leq E(\boldsymbol{x}) \tag{11c}$$

In order to obtain the tightest bound, we need to solve the following maximization problem:

$$\max_{\boldsymbol{\theta}} \Phi(\boldsymbol{\theta}) \tag{12}$$

This problem is dual to (9b). Function $\Phi(\boldsymbol{\theta})$ is concave; similar to [17,18,19], we use a subgradient method to maximize it. In order to compute a subgradient for a given vector $\boldsymbol{\theta}$, we need to solve $1 + B$ minimization subproblems in (11b). The first subproblem requires minimizing a submodular energy with pairwise terms, which can be efficiently done using graph cuts. Solving subproblems for bins b is straightforward.

It remains to specify how to choose a primal solution $\boldsymbol{x}$. Let $\boldsymbol{x}^t$ be a minimizer of the first subproblem in (11b) at step t of the subgradient method. Among labelings $\boldsymbol{x}^t$, we choose the solution with the minimum cost $E(\boldsymbol{x}^t)$.

Dual decompositions for model D. We obtained a lower bound by relaxing constraints $H_k = \sum_p x_k$ and using the fact that $\overline{H}_k \equiv |\mathcal{P}_k| - H_k$. Details are very similar to those in [12].

4 Experimental Results

In this section we describe the experimental results. We start by giving details on the setup used to compare the different models. In section 4.1, we compare the performance of the different optimization methods, and in section 4.2, using the best optimization procedure for each model, we compare the performance and robustness of such models.

Dataset. Given the difficulty in acquiring ground truth data for the cosegmentation problem, we used composites of 40 different backgrounds with 20 foreground objects from the database in [20], for which high quality alpha mattes are available. The database in [20] has more than 20 images; we selected

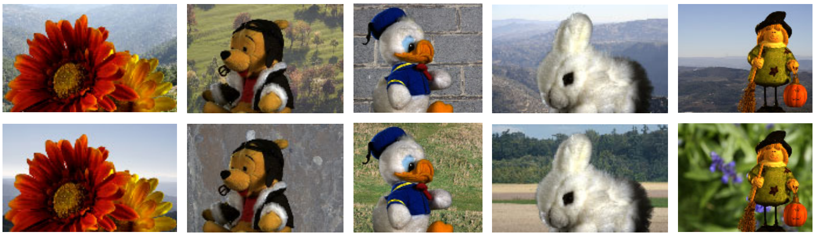

Fig. 2. Some of the images in the dataset. These images are composites using the same foreground.

objects with fewer transparencies. Representative images out of these 20 pairs are shown in Fig. 2.

We resized the images so that their maximum side is 150 pixels. Some of the models and optimization methods discussed are limited to small images, in particular, model C and QPBO. Both these optimization methods require the construction of graphs that grow quadratically with the size of the image.

The use of exactly the same foreground object in both images ensures that the histograms over pure foreground pixels match. The choice of such simplified dataset is justified by the intuition that if the models and optimization methods fail in this scenario, they will also fail in a realistic scenario where the foreground histograms may differ. In section 4.2 we also test more realistic scenarios by varying the size and illumination in one of the images.

Choosing weights μ and λ. The choice of weights for the different terms in the energy greatly affects the performance of the methods. We test the different models with different combinations of these weights. In order to reduce the search space, we fix $\lambda_i = 1$ and $\lambda_c = 50$ for all methods, similar to what is done in [1]. As for parameters λ and μ, we used leave-one-out cross-validation for each model, where parameters are allowed to take values in a discrete domain[1]. Results are given in section 4.2. For comparison, we also report results when the weights for each image are chosen optimally according to GT.

In section 4.1 we are only interested in comparing optimization methods, so we fix the weights in an ad-hoc way. For model A, we choose $\lambda = 5$ and $\mu = -2$, for model B, $\lambda = 2$ and $\mu = -10$ and for model D, $\lambda = 1$.

Histograms. We use histograms over RGB colors, using 16 bins for each color channel. Note that, in previous papers where some of the models were introduced,

[1] For model A and model B we test 16 different configurations, where $\lambda \in \{0.01, 0.1, 1, 10\}$ and $\mu \in \{-0.01, -0.1, -1, -10\}$. Since some of these configurations lead to trivial solutions, we handpick 8 other intermediate configurations that look more promising. Thus, there are 24 possible combinations of weights.

Model C allows the use of parametric maxflow for parameter learning. Fixing λ, we efficiently compute solutions for all possible values of μ using parametric maxflow. We test 4 different values for λ: 0.001, 0.01, 0.1 and 1.

Model D only has one free parameter, λ, and we test 12 different values for this weight: 0.01, 0.1, 0.5, 1, 2, 5, 10, 15, 20, 30, 40 and 100.

other appearance features were used [1,3]. Since our dataset is constructed such that the foreground histograms over color are very similar, we expect that none of the models is negatively affected by this choice of histogram quantization.

4.1 Results Comparison for Optimization Algorithms

Here we compare the optimization methods reviewed in section 3. We start by comparing Dual Decomposition with TRGC for models A and B. Since TRGC is an iterative method that requires as input an initial segmentation, we test this method with three different starting points. First, we use the solution of DD as a starting point. The second starting point is a random segmentation whose foreground histogram is constructed by having each bin take the minimum value over the corresponding bins in the full histogram of both images, i.e., $h_b = \min(|\mathcal{P}_{1b}|, |\mathcal{P}_{2b}|)$. Third, we initialize TRGC with the ground truth (GT). GT is not available at test time, and we report results only for comparison.

The results for model A are shown in the first part of Table 1. Note that in [1], where TRGC was proposed, DD was not used as a starting point. For this model, the difference between TRGC-DD and DD is very small, since TRGC starting with DD only improves the energy for two images.

Table 1. Comparison of optimization methods for Models A and B. We compare TRGC (using 3 different initial solutions), Dual Decomposition, and QPBO (only for model B). For each model, the first row shows for how many images each method gives the best energy. The second row is the gap between the energy and the lower bound (LB) obtained by DD. The values are normalized: first we add a constant to each term of the energy so that the minimum of each term becomes 0, and then scale the energy so that the lower bound corresponds to 100. The last row is the error rate: percentage of misclassified pixels over the total number of pixels.

		TRGC			DD	QPBO
		From DD	From hist	From GT		
Model A	Best energy: # cases	20	0	0	18	-
	Distance from LB	100.24	106.5	101.15	100.24	-
	Error rate	3.7%	8.1%	3.2%	3.7%	-
Model B	Best energy: # cases	13	0	7	3	0
	Distance from LB	101.59	107.56	101.77	104.20	197.29
	Error rate	3.93%	5.96%	2.85%	3.92%	51.77%

The results for model B are shown in the second part of Table 1. Although QPBO also provides a lower bound, we used the lower bound obtained by Dual Decomposition since in our experiments, it was always better than the one provided by QPBO.

We conclude that a combination of DD and TRGC, using DD solution as a starting point for TRGC, is the best performing method for both model A and B, and this is the method used in the next section for model comparison.

Surprisingly, the performance obtained for the QPBO method contrasts with the one reported in [2], since for this experiment the number of pixels left unlabeled by this method was 90%. Note that in [2], the authors used a different spatially varying unary term which may induce differences. They also report that the performance of the method deteriorates when weight λ is increased. In the case considered, where w_p is constant, small values of λ lead to trivial solutions.

In order to better understand why QPBO fails, we ran the method with a fixed ballooning force, $\mu = -10$, and different values of λ. In Table 2, we show the percentage of pixels that were labeled one, zero, or left unlabeled. For intermediate values of λ, the number of unlabeled pixels is more than 90%. For such values, QPBO is not reliable as an optimization method. On the other hand, for extreme values of λ, QPBO labels more pixels, but the resulting model is not meaningful, for example, for the case $\lambda = 10^{-3}$, all pixels for all images considered were labeled 1.

Table 2. QPBO results. Percentage of pixels labeled 1, 0 or left unlabeled by the QPBO method for different values of weight λ.

λ	10^{-3}	10^{-2}	10^{-1}	10^{0}	10^{1}	10^{2}	10^{3}
Labeled 1	100	64.49	9.52	0.18	0.03	0.03	0.03
Labeled 0	0	0	0	0	22.68	25.66	24.22
Unlabeled	0	35.51	90.48	99.82	77.30	74.31	75.75

Dual Decomposition for model D. We compared two different optimization methods for model D: the EM-style iterative procedure of [11] and a DD approach. For the EM-style optimization, we initialized the color models in the same way as discussed before for TRGC initialization when taking the histograms' intersection. Since DD provides a lower bound, we compared the gap between the lower bound and the energy obtained by both models. For DD this gap is 109.5 and for the EM-style optimization it is 103.4. The average gap is reduced to 103.2, if the best method is chosen for each image. This is very similar to the gap obtained by the iterative technique and we conclude that the improvement of using DD is only marginal for this problem and we report results using the EM-style optimization.

4.2 Results Comparison for Models

In this section we compare the four different models. We present results for three different cases. In the first case, we use the original images (some of the images are shown in Fig. 2), where the same foreground is composed with two different backgrounds. This is the simplest case and the error rate is reported in the first row of Table 3.

In the second case, we consider images of different sizes, reducing one of the images to 90% and 80% of the original size. This leads to a more complicated cosegmentation problem, where the object has different sizes in both images.

In the third case, in order to simulate illumination changes, we add a constant to all RGB values (ranging from 0 to 255) of one of the images. We show results for two different values of this constant: 3 and 6.

In Table 3, we also present the histogram similarity for the different cases. This similarity is given by: $100 - 100 \times \frac{\sum_b |h_{1b}^{GT} - h_{2b}^{GT}|}{\sum_b h_{1b}^{GT} + h_{2b}^{GT}}$ where h_k^{GT} is the histogram of image k computed over foreground ground truth pixels. This similarity can be seen as a rough measure of the difficulty of the problem, and the higher it is, the simpler the problem.

Table 3. Error rate using leave-one-out cross-validation. We compare the error rate for the different methods in 3 different scenarios. We also report the standard error of estimating the mean of the error rate. For the first case we use the original composites. In the second case we consider images of different sizes, reducing one of the images to 90% and 80% of the original size. In the third case, in order to simulate illumination changes, we add a constant to all RGB values of one of the images, 3 and 6. The last column shows the similarity of the foreground histograms of both images.

	Model A	Model B	Model C	Model D	Histogram similarity
Original images	4.6% ±0.8	3.9% ±0.7	22.0% ±3.9	4.3% ±0.3	93.4
Resized to 90%	4.7% ±0.4	5.7% ±0.8	16.3% ±2.4	4.9% ±0.5	84.6
Resized to 80%	7.8% ±1.3	9.7% ±1.4	17.4% ±3.0	5.1% ±1.0	74.2
RGB +3	4.4% ±0.4	7.1% ±1.1	21.4% ±4.3	3.7% ±0.3	84.6
RGB +6	5.5% ±0.5	12.3% ±1.7	20.3% ±2.5	4.0% ±0.4	76.3

From the results presented in Table 3 we take the following statistically significant observations:

- Models A, B, and D perform similarly for the simplest case.
- Model C is the worst performing model since it produces in every case considerably higher error rates.
- Model D is the most robust to changes in size and illumination.
- Comparing both models based on histogram distances, the L1-norm (Model A) is more robust than the L2-norm (Model B), for the cases where there are small variations of foreground.

Some methods may be affected negatively by the way the weighting parameters are chosen, since image measurements are not taken into account. In order to fairly compare the methods without introducing this type of bias, we also present results in Table 4 for the case where the weights λ and μ are chosen independently for each image, so that the error rate is minimized.

Table 4. Error rate without cross validation. These results correspond to choosing the best weights λ and μ according to GT for each image individually. They should be compared with Table 3.

	Model A	Model B	Model C	Model D	Histogram similarity
Original images	3.2% ±0.3	2.9% ±0.3	8.8% ±1.9	3.2% ±0.3	93.4
Resized to 90%	4.2% ±0.4	4.0% ±0.4	8.1% ±1.7	3.2% ±0.3	84.6
Resized to 80%	5.2% ±0.6	6.2% ±0.6	7.0% ±1.4	3.2% ±0.3	74.2
RGB +3	3.3% ±0.3	4.0% ±0.2	9.3% ±1.8	3.2% ±0.2	84.6
RGB +6	4.3% ±0.4	8.0% ±1.2	9.2% ±1.8	3.3% ±0.2	76.3

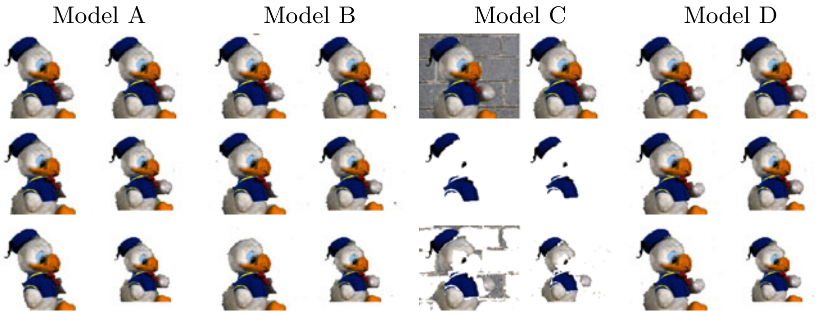

Fig. 3. Results without cross-validation. Segmentation obtained for each model when reducing the size of the second image.

Comparing tables 3 and 4, it can be seen that model C has the greatest improvement in error rate when the choice of weights is done independently for each image. However, it still remains the worst performing model.

In Fig. 3, we show some cosegmentation results for a pair of images for different sizes of the second image. The results shown agree with the insights discussed in Sect. 2.2. When the size of the images differ, both models A and B incorrectly cut some parts of the object, in order to improve the matching of the resulting foreground histograms. Model C gives unpredictable results due to the mentioned bias. Model D copes better with the changes in image size.

4.3 Results for Real Images

Following a reviewer's suggestion, we tested the different models on the real images used in [3][2].

[2] Images and GT are available from http://www.cs.wisc.edu/˜vsingh/pairimages.tar.gz. We chose 20 pairs of images from this dataset, excluding the ones which were created in a similar way to our dataset.

We observed that the histogram quantization used in the rest of the paper is not appropriate for these images, since there are significant differences in the foreground color histograms and the overlap of the background color histograms is large. The overlap for the foreground histograms is 39% which is considerably lower than the overlap reported in the last column of Table 3. On the other hand, the overlap of background histograms is 21% compared to 8% for our dataset of composed images. This affected the results negatively and the error rates are between 20% and 30% for all image pairs. The use of better histogram quantization would considerably improve the performance for all methods.

This observation further supports our use of composed images, since the goal of the paper is to compare the performance of the different methods in a scenario where external factors with a negative impact could be easily controlled.

Note that, the results reported for the same images in [2,3] used user interaction and the results in [1] used various features to calculate the histograms.

5 Conclusions and Future Work

Recently, several models for cosegmentation have been proposed some of which lead to challenging optimization problems. We showed that they are outperformed by a natural extension of the Boykov-Jolly model, which has not been considered in the context of cosegmentation before. The improvement of model D over models B and especially C is substantial. The gap between models D and A is less significant, and potentially could be affected by the choice of a dataset. However, model D has two clear advantages: it has one less parameter, and it allows the use of an effective and fast EM-style optimization.

To enable a fair comparison of models, we had to improve on optimization techniques in [1,2]. We believe the Dual Decomposition method that we used for models A and B was adequate for our task. Although, we did not get verifiable global minima, the gap between the lower bound and the energy was small enough, and furthermore, using ground truth to initialize an iterative technique (TRGC) led to higher energies compared to DD.

In the future, we plan to gather a larger and more challenging dataset of images for cosegmentation, including the ones used in Sect. 4.3. The focus will be on the construction of discriminative histograms, that take into account not only color but also other features like SIFT and Gabor filters as in [8].

Acknowledgements. We thank Vikas Singh for answering questions about his implementation.

References

1. Rother, C., Kolmogorov, V., Minka, T., Blake, A.: Cosegmentation of image pairs by histogram matching - incorporating a global constraint into MRFs. In: CVPR (2006)
2. Mukherjee, L., Singh, V., Dyer, C.R.: Half-integrality based algorithms for cosegmentation of images. In: CVPR (2009)

3. Hochbaum, D.S., Singh, V.: An efficient algorithm for co-segmentation. In: ICCV (2009)
4. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In: ICCV (2001)
5. Cui, J., Yang, Q., Wen, F., Wu, Q., Zhang, C., Cool, L.V., Tang, X.: Transductive object cutout. In: CVPR (2008)
6. Batra, D., Kowdle, A., Parikh, D., Luo, J., Chen, T.: iCoseg: Interactive cosegmentation with intelligent scribble guidance. In: CVPR (2010)
7. Winn, J., Jojic, N.: Locus: learning object classes with unsupervised segmentation. In: ICCV (2005)
8. Joulin, A., Bach, F., Ponce, J.: Discriminative clustering for image co-segmentation. In: CVPR (2010)
9. Batra, D., Parikh, D., Kowdle, A., Chen, T., Luo, J.: Seed image selection in interactive cosegmentation. In: ICIP (2009)
10. Personal communication with Vikas Singh
11. Rother, C., Kolmogorov, V., Blake, A.: Grabcut - interactive foreground extraction using iterated graph cuts. In: SIGGRAPH (2004)
12. Vicente, S., Kolmogorov, V., Rother, C.: Joint optimization of segmentation and appearance models. In: ICCV (2009)
13. Hammer, P.L., Hansen, P., Simeone, B.: Roof duality, complementation and persistency in quadratic 0-1 optimization. Math. Programming 28, 121–155 (1984)
14. Boros, E., Hammer, P.L.: Pseudo-boolean optimization. Discrete Applied Mathematics 123(1-3), 155–225 (2002)
15. Bertsekas, D.: Nonlinear Programming. Athena Scientific, Belmont(1999)
16. Wainwright, M., Jaakkola, T., Willsky, A.: MAP estimation via agreement on trees: Message-passing and linear-programming approaches. IEEE Trans. Information Theory 51(11), 3697–3717 (2005)
17. Schlesinger, M.I., Giginyak, V.V.: Solution to structural recognition (MAX,+)-problems by their equivalent transformations. Part 1. In: Control Systems and Computers, pp. 3–15 (2007)
18. Schlesinger, M.I., Giginyak, V.V.: Solution to structural recognition (MAX,+)-problems by their equivalent transformations. Part 2. In: Control Systems and Computers, pp. 3–18 (2007)
19. Komodakis, N., Paragios, N., Tziritas, G.: MRF optimization via dual decomposition: Message-passing revisited. In: ICCV (2005)
20. Rhemann, C., Rother, C., Wang, J., Gelautz, M., Kohli, P., Rott, P.: A perceptually motivated online benchmark for image matting. In: CVPR (2009)