

# An efficient algorithm for Co-segmentation

Dorit S. Hochbaum

[www.ieor.berkeley.edu/~hochbaum](http://www.ieor.berkeley.edu/~hochbaum)

Haas School of Business, and  
Industrial Eng. and Operations Research  
Univ. of California, Berkeley

Vikas Singh

[www.biostat.wisc.edu/~vsingh](http://www.biostat.wisc.edu/~vsingh)

Biostatistics & Medical Informatics,  
and Computer Sciences  
Univ. of Wisconsin–Madison

## Abstract

*This paper is focused on the Co-segmentation problem [1] – where the objective is to segment a similar object from a pair of images. The background in the two images may be arbitrary; therefore, simultaneous segmentation of both images must be performed with a requirement that the appearance of the two sets of foreground pixels in the respective images are consistent. Existing approaches [1, 2] cast this problem as a Markov Random Field (MRF) based segmentation of the image pair with a regularized difference of the two histograms – assuming a Gaussian prior on the foreground appearance [1] or by calculating the sum of squared differences [2]. Both are interesting formulations but lead to difficult optimization problems, due to the presence of the second (histogram difference) term. The model proposed here bypasses measurement of the histogram differences in a direct fashion; we show that this enables obtaining efficient solutions to the underlying optimization model. Our new algorithm is similar to the existing methods in spirit, but differs substantially in that it can be solved to optimality in polynomial time using a maximum flow procedure on an appropriately constructed graph. We discuss our ideas and present promising experimental results.*

## 1. Introduction

The idea of co-segmentation, first introduced in [1], refers to the simultaneous segmentation of two images. The problem is well illustrated by the example in Fig. 1, where the same (or similar) object appears in two different images, and we seek to perform a segmentation of only the similar regions in both views. This problem was partly motivated in [1] by the need for computing meaningful similarity measures between images of the same subject but with different (and unrelated) backdrops in image retrieval applications [3]. A related goal was to facilitate segmentation of an object (or a region of interest) by providing minimal additional information (such as just one additional image). The

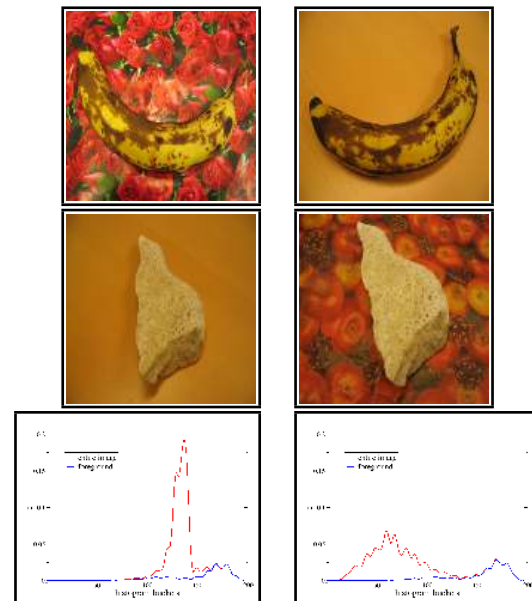


Figure 1. A similar object in two images in rows 1-2. The histogram of the foreground (of row 2 images) is shown in row 3.

idea has been utilized in a number of other *concurrent* foreground extraction tasks using multiple images [4], images acquired with/without camera flash [5], image sequences [6], and for identifying individuals using image collections [7]. Later in the paper, we discuss how the idea may be applied for pathology identification problems in biomedical images. The purpose of this paper, however, is to investigate efficient means of solving (the underlying optimization problem of) co-segmentation.

The identification of similar objects in more than one image is a fundamental problem in computer vision and has relied on user annotation or construction of models [8, 9]. A number of recent techniques [1, 10, 11, 12], however, have preferred an unsupervised (or semi-supervised) approach to the problem and obtained good overall performance. Co-segmentation belongs to this second category. The key idea adopted in [1] was to apply a MRF segmentation on both images with an additional term that penalizes the variation in the histograms of the foreground regions in the two im-

ages, see Fig. 1. The energy function is expressed as

$$\min \quad E(\mathbf{x}_1, \mathbf{x}_2) = E_1(\mathbf{x}_1, \mathbf{x}_2) + E^{\text{global}}(\hat{h}_1, \hat{h}_2), \quad (1)$$

where  $\mathbf{x}_1, \mathbf{x}_2 \in \{0, 1\}$  are binary variables indicating the assignment of pixels to the background or foreground,  $E_1(\cdot, \cdot)$  denotes the sum of the image-wise MRF energies, and  $E^{\text{global}}(\hat{h}_1, \hat{h}_2)$  measures the difference between a Gaussian model of the two foreground histograms,  $\hat{h}_1$  and  $\hat{h}_2$ . The authors [1, 2] note that efficient optimization of (1) is not tractable. Therefore, the proposed procedure initializes the segmentations, and incrementally improves one of the segmentations keeping the other fixed (and vice-versa) until no further improvements are possible. In subsequent work [6], a generative model was proposed for performing co-segmentation in image sequences, and a locally maximal marginal log posterior estimate was obtained using an expectation maximization (EM) algorithm with certain convergence criteria. Later, the authors in [4] extended many of these ideas further by incorporating local context, i.e., patterns characterizing the local color and edge configurations. This led to improved results relative to [1, 13]. However, the technique [4] was focused on the empirical performance, and less effort was devoted toward better means of optimizing the co-segmentation cost function in [1]. Recently, [2] proposes addressing some of these difficulties by replacing the second term with the squared difference of the two histograms. This approach no longer requires the histograms to be Gaussian, and leads to a quadratic pseudoboollean optimization model [14]. The authors prove that their formulation yields half-integral solutions (i.e.,  $\{0, \frac{1}{2}, 1\}$ ) to the optimization problem. However, the problem still remains hard (and cannot be solved optimally), and obtaining provably good quality guarantees in a general case is difficult. On the practical side, solving the linear program in [2] resulting from large images may be computationally intensive.

**Carrot or Stick?** The simplest interpretation of the co-segmentation model, as discussed above, is that it encourages good and coherent segmentation of both images with an additional requirement of consistency between foreground histograms. To enforce this requirement, variations between foreground histograms are *penalized* in [1, 4, 2], but this leads to intractable optimization problems. On the other hand, notice that a similar effect may also be achieved by *rewarding* consistency in the two foreground histograms (rather than explicitly penalizing their difference): *the carrot or the stick*<sup>1</sup>. While this seems logical once we choose a suitable measure of histogram consistency, the key question is: what is the benefit of adopting this second approach? In the following sections, we will show that this modification leads to a *polynomial time algorithm* for co-segmentation. This is the primary contribution of this paper.

<sup>1</sup>a mechanism of offering rewards (e.g., carrot) or threatening punishment (e.g., stick) to induce a desired behavior.

## 2. Preliminaries

In the co-segmentation setting, we are given two images for segmentation:  $I^{(1)}$  and  $I^{(2)}$ . The images are of the same size consisting of  $n$  pixels each, where the  $j^{\text{th}}$  pixel in the  $q^{\text{th}}$  image is denoted by  $I_j^{(q)}$ , for  $j = 1, \dots, n$  and  $q = 1, 2$ . We are also given a classification of each pixel in each image into one of  $K$  ‘buckets’ in a histogram for each image. Let the histogram buckets (each bucket corresponds to an intensity range) be given as  $h_1, h_2, \dots, h_K$ . For each image  $I^{(q)}$ ,  $q = 1, 2$ , this may be specified in terms of a matrix  $\mathbf{B}^{(q)}$  of size  $n \times K$  such that for pixel  $j$  and bucket  $h_k$ ,

$$B_{j,k}^{(q)} = \begin{cases} 1 & \text{if } I_j^{(q)} \in H_k; \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

That is, the entry  $B_{j,k}^{(q)}$  is 1 if the intensity of pixel  $I_j^{(q)}$  falls in the intensity range of bucket  $h_k$ , where  $q$  refers to either the first or the second image.

A segmentation of each image will partition the set of pixels into *foreground* versus *background* pixels. Our interest is to ensure that the foreground in the two images are similar. Toward this goal, the objective is to get (1) the number of pixels that are in the foreground, and (2) the number of pixels in  $H_k$ , to be approximately similar in both images. One strategy is to define *similarity* between all pairs of pixels  $I_i^{(1)}$  and  $I_j^{(2)}$ . We can say that the pair  $i, j$  is similar if both belong to  $H_k$  and designate a similarity weight  $s_{ij}$  to be equal to 1 if that happens. Formally,

$$s_{ij} = \begin{cases} 1 & \text{if } \exists k \text{ such that } B_{i,k}^{(1)} = B_{j,k}^{(2)} = 1 \\ 0 & \text{otherwise.} \end{cases}$$

Let  $x_j^{(q)}$  be a binary variable indicating whether pixel  $I_j^{(q)}$  is classified in the foreground:

$$x_j^{(q)} = \begin{cases} 1 & \text{if } I_j^{(q)} \text{ is classified as foreground} \\ 0 & \text{if } I_j^{(q)} \text{ is classified as background.} \end{cases}$$

The number of pixels in the foreground of  $I^{(1)}$  that belong to  $H_k$  is denoted by  $a_k = \sum_{j=1}^n B_{j,k}^{(1)} x_j^{(1)}$  and the number in  $I^{(2)}$  that belong to  $H_k$  is denoted by  $b_k = \sum_{j=1}^n B_{j,k}^{(2)} x_j^{(2)}$ . Let the total number of foreground pixels in  $I^{(1)}$  and  $I^{(2)}$  be  $F_1$  and  $F_2$  respectively. We model a measure of similarity of the two foreground features as the optimal solution to

$$\begin{aligned} \max \quad & \sum_{k=1}^K a_k b_k \\ \text{subject to} \quad & \sum_{k=1}^K a_k = |F_1| \\ & \sum_{k=1}^K b_k = |F_2|. \end{aligned} \quad (3)$$

Our rationale is that for each pixel  $p$  assigned as foreground in the first image, we offer a reward for also selecting (as part of the foreground in the second image) a pixel  $q$  which is similar to  $p$ . Similarity which is specified by a binary

$s_{pq}$  depends on whether  $p$  and  $q$  belong to the same bucket, and may be allowed to vary in  $[0, 1]$  as a function of the likelihood of the match ( $p \rightarrow q$ ) detected by some feature extraction method. For  $|F_1|, |F_2|$  fixed, the optimization process seeks to maximize the number of pixel pairs (one from each image) with identical histogram buckets. We note that treating  $|F_1| = |F_2|$  as normalization constants for  $a$  and  $b$  resp., this is also similar to Hellinger affinity (see [15], pg. 24), frequently used in computer vision [16].

### 3. Problem Statement

Maximizing similarity of histograms as in (3), by itself is not sufficient to obtain meaningful segmentations. This is because Co-segmentation must take the spatial homogeneity of the images into account also. This may be achieved by introducing the adjacency relationship between neighboring pixels as an additional bias into the maximization in (3). Another option, which we adopt here, is to segment both images while using the similarity in (3) as a bias term.

#### 3.1. MRF segmentation

We formulate the task of segmenting both images as a binary labeling of Markov Random Field (MRF) on the graphs corresponding to the input images [17, 18]. That is, in each image  $I$ , we find the assignment of values to every pixel, as either foreground or background label. This is represented by a binary variable  $x_j$  assigned to each pixel  $j$  and is equal to 1 if the pixel is assigned to the foreground. The assignment is such that the total deviation and separation penalties are minimized. The *deviation (or data) penalty*,  $d_j$ , is charged for a pixel that is set in the foreground, although there is a-priori information indicating it should be in the background. The *separation or smoothness* penalty  $w_{pq}$  measures the cost of assigning different labels to two neighboring pixels,  $p \sim q$ . As in [7, 4], we give  $w_{pq}$  as  $\exp(-\beta\|p - q\|^2)$ , where  $\beta$  is a constant. The MRF formulation for one image is then:

$$\begin{aligned} \min \quad & \sum d_j x_j + \sum_{i \sim j} w_{ij} y_{ij} \\ \text{subject to} \quad & x_i - x_j \leq y_{ij} \\ & x_j - x_i \leq y_{ji}. \\ & x_i, y_{ij} \text{ binary for } i, j = 1, \dots, n. \end{aligned} \quad (4)$$

#### 3.2. Co-segmentation

Our model attempts to simultaneously minimize the separation and deviation terms in the MRF model for *each* image as well as maximize the similarity (rather than minimize the difference [1, 2]) between the foreground features in the two images as specified in (3). As these are two conflicting and incompatible goals, we use a linear combination of the two objectives (treating the second term as a bias). Let  $\lambda$

be a coefficient expressing the relative weights of the two objectives [1, 2]: when the value of  $\lambda$  is high, then similarity is the most important requirement, and when it is low, the MRF penalties are dominant. Let  $z_{ij}$  be a variable equal to 1 if  $I_i^{(1)} \in F_1$  and  $I_j^{(2)} \in F_2$ . Our objective function minimizes a combination of the penalties incurred by the MRF optimization in each image, and *subtracts* the similarity measure of the number of pairs of the same histogram buckets in the resulting two foreground features. Since we have a minimization in (4), a high similarity in the foreground features serves as a reward, exactly as desired.

In this formulation, we seek an assignment of the pixel to the foreground or the background. So we may simplify the notation:  $d_j^{(1)}, d_j^{(2)}$  are the deviation penalties charged for placing pixel  $j$  in the foreground of image 1 and 2 respectively. These penalties can be positive or negative. The minimization objective includes terms representing the MRF optimization in both images,

$$\sum d_j^{(1)} x_j^{(1)} + \sum_{i \sim j} w_{ij} y_{ij}^{(1)} + \sum d_j^{(2)} x_j^{(2)} + \sum_{i \sim j} w_{ij} y_{ij}^{(2)}.$$

Simultaneously, we also wish to maximize the benefit of high similarities between corresponding histogram buckets in both images represented as:

$$\sum_{i \in I^{(1)}, j \in I^{(2)}} s_{ij} z_{ij}.$$

Since  $s_{ij}$  is equal to 1 only for “matching” histogram buckets, this latter term can also be written as:

$$\sum_{k=1}^K \sum_{i \in I^{(1)} \cap H_k, j \in I^{(2)} \cap H_k} z_{ij}. \quad (5)$$

Notice that (5) is equivalent to the requirement specified in (3). Our formulation of the co-segmentation problem is then a linear combination of the MRF minimization and similarity maximization objectives as follows:

$$\begin{aligned} \min \quad & \sum d_j^{(1)} x_j^{(1)} + \sum_{i \sim j} w_{ij} y_{ij}^{(1)} + \sum d_j^{(2)} x_j^{(2)} \\ & + \sum_{i \sim j} w_{ij} y_{ij}^{(2)} - \lambda \sum_{k=1}^K \sum_{i \in I^{(1)} \cap H_k, j \in I^{(2)} \cap H_k} z_{ij} \\ \text{subject to} \quad & z_{ij} \leq x_i^{(1)} \text{ for } i \in I^{(1)} \\ & z_{ij} \leq x_j^{(2)} \text{ for } j \in I^{(2)} \\ \text{(Co-seg)} \quad & x_i^{(q)} - x_j^{(q)} \leq y_{ij}^{(q)} \text{ for } q = 1, 2 \\ & x_j^{(q)} - x_i^{(q)} \leq y_{ji}^{(q)} \text{ for } q = 1, 2 \\ & x_i^{(q)}, y_{ij}^{(q)}, z_{ij} \text{ binary for } q = 1, 2, \\ & i, j = 1, \dots, n. \end{aligned}$$

**Correctness.** To verify correctness, observe that the first set of constraints on  $z_j$  ensures that the binary variable  $z_j$  can be equal to 1 only if both pixels  $i$  and  $j$ , in the first and second images respectively, are selected in the foreground.

The second set of constraints is to guarantee that if adjacent pixels  $i$  and  $j$  in one of the images are assigned such that one is in the foreground and the other is in the background, then the separation penalty for that neighboring pair is charged. Notice that we make use of the objective that drives  $z_{ij}$  to be as large as possible (that is 1) and  $y_{ij}$  to be as small as possible (that is 0). It is not difficult to verify that:

**Property 3.1** *The model of the (Co-seg) problem is defined on monotone constraints [19] and with a totally unimodular constraint matrix.*

Due to Property 3.1, we can make use of a construction of an  $s, t$  graph  $G$ , where the solution to the  $s, t$ -cut problem will provide an *optimal* solution to the (Co-seg) problem.

#### 4. The graph construction

We now show the construction of the  $s, t$  graph  $G$  which will be used to solve (Co-seg): For each of the two images, the graph contains a grid of nodes, called here pixel-nodes, one corresponding to each pixel. To achieve *only* the MRF segmentation for both images specified as (4), we can use a graph construction similar to the one described in [17], with either the 4-neighbor or the 8-neighbor or any other form of neighborhood topology used to describe the adjacency relationship between pixel-nodes. But to make it suitable for co-segmentation, the graph will be modified, details of which will be described shortly.

We denote the pixel-nodes in the graph by  $V_x$ , as each corresponds to a variable  $x_i$ . The graph  $G$  contains the “dummy” nodes  $s$  and  $t$ . Each pixel-node  $j$  has a weight  $d_j$  associated with it, as shown in (4). If  $d_j > 0$ , then there is an arc  $(j, t)$  of capacity  $d_j$ . If  $d_j < 0$ , then there is an arc  $(s, j)$  of capacity  $-d_j$ . We partition  $V_x$  to  $V_{x+} \cup V_{x-} \cup V_0$ , where for each node  $j$  in  $V_{x+}$ ,  $d_j > 0$ , and for each node  $j$  in  $V_{x-}$ ,  $d_j < 0$ . For each pair of adjacent nodes  $i$  and  $j$  there is a capacity  $w_{ij}$  on both directed arcs  $(i, j)$  and  $(j, i)$ .

We now outline the key modifications. In addition to the nodes for the pixels in the two images, there is a *similarity node*, or  $z$ -node, for each pair  $(i_k^1, i_k^2)$  so that  $i_k^1 \in I^{(1)} \cap H_k$  and  $i_k^2 \in I^{(2)} \cap H_k$ . This node corresponds to the variable  $z_{i_k^1, i_k^2}$  in the (Co-seg) model. We denote the set of similarity nodes by  $V_z$ , and link each such node to both  $i_k^1$  and  $i_k^2$  with arcs of infinite capacity. We then link this node to the source with an arc  $(s, (i_k^1, i_k^2))$  of capacity  $\lambda$  (weight of the bias).

The constructed graph is  $G = (V \cup \{s, t\}, A)$  with  $V = V_x \cup V_z$  and  $A$  the set of arcs. The set of arcs  $A$  is the union of: the set of adjacency arcs in  $I_1, A_1$ ; the set of adjacency arcs in  $I_2, A_2$ ; the set of arcs  $(j, t)$  directed to the sink from all nodes  $j \in V_{x+}$ ; the set of arcs  $(s, j)$  from the source to all nodes  $j \in V_{x-}$ ; one arc  $(s, z)$  for each node  $z \in V_z$  and two arcs from each  $z$  to the respective pixel nodes. An illustration of the graph is shown in Figure 2.

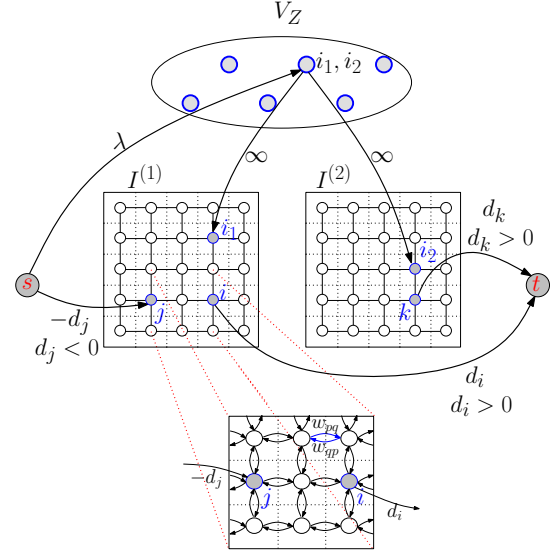


Figure 2. The construction of the graph  $G$  with two dummy nodes, the set of pixels in the two images  $I^{(1)}$  and  $I^{(2)}$ , and the set of similarity nodes  $V_z$ . Some nodes and arcs are annotated to show the graph structure.

#### 5. The algorithm (Co-Seg)

For a finite cut  $(S \cup \{s\}, T \cup \{t\})$  of  $G$ , we refer to the set of nodes in  $V_{x+} \cap S$  as  $S_{x+}$  and we let  $S_{x-} = V_{x-} \cap S$  and  $S_z = V_z \cap S$ . The analogous notation is used for those sets intersecting  $T$ . We can now show the following result.

**Theorem 5.1** *Let  $(S \cup \{s\}, T \cup \{t\})$  be the minimum  $s, t$ -cut in the graph  $G$  obtained using a max-flow algorithm. Then the optimal solution to (Co-Seg) is achieved by setting  $x_i = 1$  for each pixel node in the source set  $S$  and every  $z_{i_k^1, i_k^2} = 1$  for each similarity node in the sink set  $T$ .*

**Proof:** The graph  $G$  has at least one finite capacity cut,  $(\{s\}, V \cup \{t\})$ . Let  $(S \cup \{s\}, T \cup \{t\})$  be a partition of  $V \cup \{s, t\}$  forming a finite  $s, t$ -cut in  $G$ . Such a cut corresponds to a *feasible* solution since if  $z_{i_k^1, i_k^2} = 1$  then also  $x_{i_k^1} = 1$  and  $x_{i_k^2} = 1$ , as otherwise an infinite capacity arc will contribute to the capacity of the cut which violates its finiteness. Other than satisfying this constraint any setting of the values of the variables  $x$  is feasible. The values of the variables  $y_{ij}^{(q)}$  are determined so they are  $= 1$  if the respective arc in the graph is directed from a node in the source set ( $S$  node) to a node in the sink set ( $T$  node). Therefore, the constraints for the  $y$  variables are satisfied as well. This shows that a finite cut corresponds to a feasible solution to the problem (Co-Seg). We now compute this cut’s capacity:

$$C(S \cup \{s\}, T \cup \{t\}) = \sum_{i \in S_{x+}} d_i + \sum_{j \in T_{x-}} (-d_j) + \sum_{i \in S_x, j \in T_x} w_{ij} + \lambda |T_z|.$$

We note that

$$\sum_{j \in T_{x-}} d_j = \sum_{j \in V_x} d_j - \sum_{j \in S_{x-}} d_j$$

and also  $\lambda|T_z| = \lambda|V_z| - \lambda|S_z|$ . Therefore, the cut value is

$$C(S \cup \{s\}, T \cup \{t\}) = -\sum_{j \in V_x} d_j + \lambda|V_z| + \sum_{i \in S_x} d_i + \sum_{i \in S_x, j \in T_x} w_{ij} - \lambda|S_z|.$$

The first two terms in the sum are constant. Thus, minimizing  $C(S \cup \{s\}, T \cup \{t\})$  is equivalent to minimizing  $\sum_{i \in S_x} d_i + \sum_{i \in S_x, j \in T_x} w_{ij} - \lambda|S_z|$ , which is precisely the objective value of the (Co-Seg) problem, when setting the  $x$  and  $z$  variables with corresponding nodes in  $S$  to 1. ■

## 6. Experimental Results

In this section, we discuss our experiments for evaluating the performance of our algorithm qualitatively and relative to earlier approaches. Later, we present experiments to assess the running time of the algorithm on standard image sizes, and look at the dependence of the results on some user tunable parameters. In the experiments described here, we used histograms derived from the image intensity values and Gabor filter based texture features [20]. Our method is transparent of the underlying appearance model (i.e., parameterization of its distribution), and other texture representations [21] can be used easily, if desired. For comparisons with existing techniques, we used an implementation of the algorithm in [1]: we start with a segmentation of the two images using graph cuts, and then incrementally force the foregrounds to be consistent with one another (in an alternative fashion). This requires solving a sequence of graph cuts and the process terminates once the algorithm has converged or the number of iterations have been reached (number of iterations was set to 10).

### 6.1. Qualitative and quantitative analysis

We first present results obtained by the proposed algorithm on a set of images from [1] in Figs. 3-4. In the first pair of images (stone), we see that a graph cuts segmentation works well on the second image but on the first image the lower part of the stone is not properly segmented. The two co-segmentation algorithms, however, successfully distinguish the object from the background in both images. In the second set of images (banana), graph cuts oversegments the second image. The first image, however, is easier to segment, and this characteristic is exploited by the co-segmentation algorithms to significantly reduce the number of misclassified pixels. Similarly, the first image in the third pair is particularly difficult to segment because of negligible contrast variation between the object of interest and the background. As a result, a graph cuts segmentation does not perform satisfactorily. Co-segmentation exploits the stronger discontinuity between the object and the background to correctly segment the first image also. We continue the results in Fig. 4, where graph cuts oversegments (and undersegments) in the first (and second) image pair

respectively. Both co-segmentation algorithms can successfully identify the region of interest from background. The performance of the algorithms on the remaining five images in our dataset was similar. We found that co-segmentation improves upon the graph-cuts segmentation by 3-8%. Expectedly, the improvements are more prominent when one of the images is “easy” – this allows the process to utilize the additional information to segment an otherwise difficult second image. While there were small variations in the solution from our algorithm and [1] (see misclassification error in Figs. 3-4), these differences (w.r.t. accuracy) were not significant. In general, on image pairs suitable for co-segmentation the performance of both algorithms is comparable which provides empirical evidence (follow-up to discussion in §2- 3.2) that the proposed model is suitable for the problem. A *practical* advantage offered by our solution is that it is *non-iterative* and requires *only one* max-flow procedure (discussed in detail in Section 6.3).

#### Illumination and Scale.

Figure 5 shows a few additional images collected from image hosting websites (such as Flickr) where co-segmentation is useful. These examples illustrate that by using good histogram features, co-segmentation is relatively invariant to moderate changes in illumination. In addition, due to our choice of rewarding similarity in histogram features (see (5)), small changes in scale of the object (between images) do not have a significant impact on the empirical performance of the algorithm.

### 6.2. Dependencies and variations

#### Bias magnitude.

In Fig. 6, we illustrate the effect of the final segmentation as a response to introducing variation in magnitude of the introduced bias (to favor histogram similarity). For  $\lambda$  too small, a number of additional pixels are part of the foreground due to a strong influence of the separation penalty. For a larger  $\lambda$  value (and a large number of  $z$ -nodes, see Fig. 2), the cumulative histogram similarity reward in assigning additional pixels to the foreground outweighs the corresponding MRF penalty. In general, a “sweet spot” for  $\lambda$  depends less on the specific image, and more on the number of buckets chosen to specify the histogram (i.e., number of  $z$ -nodes). Therefore, a suitable value can be evaluated using cross-validation (in our experiments,  $\lambda = 0.001$  worked well). We note, however, that it is easy to make this procedure rigorous if desired. This involves parameterizing  $\lambda$ , solving a *single* parametric max-flow procedure [22, 23], and finding the correlation coefficient of the two foreground regions for each breakpoint.

#### Number of histogram buckets.

The number of histogram buckets should be chosen such that the corresponding similarity nodes ( $z$ -nodes) in Fig. 2 are sensitive *as well as* specific. That is, the number should



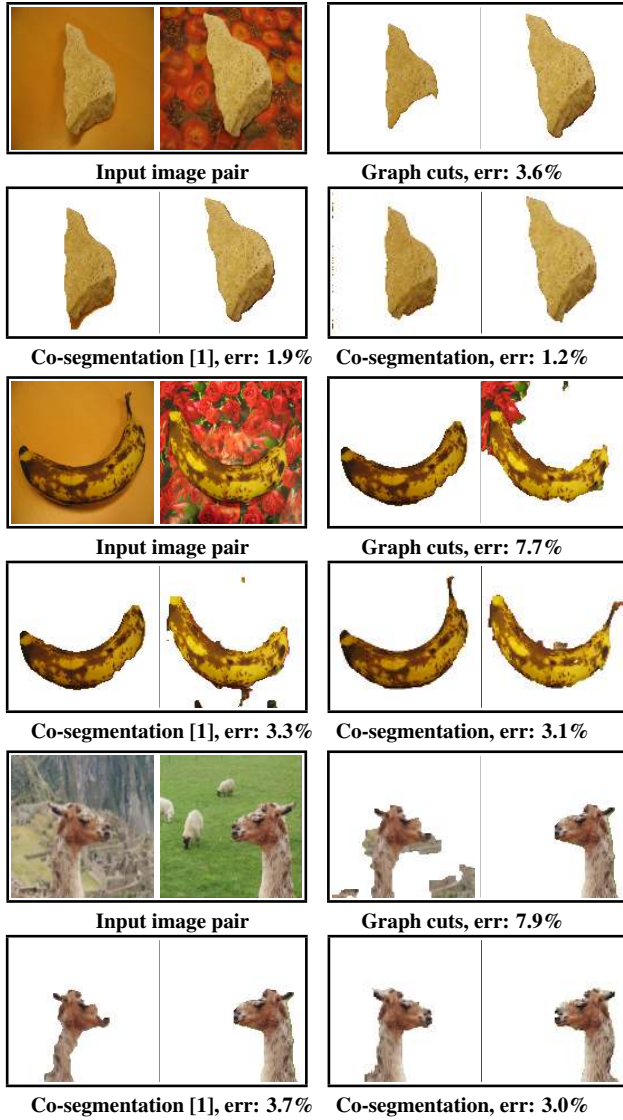


Figure 3. The first row shows the input image pair and segmentation obtained using graph cuts on each image. The second row shows segmentations obtained using the two co-segmentation algorithms: [1] (left) and our solution (right).

not be too large (very similar pixels hash to different buckets), it should also not be too small (arbitrary pixel pairs hash to the same bucket). We found that 15-30 buckets per color channel works well. Fig. 7 shows that using very few buckets leads to oversegmentation. Further, a small number of buckets yields a large set of  $z$ -nodes, which leads to an increase in running time.

### 6.3. Running time

We now demonstrate a useful advantage of our algorithm in terms of running time. Fig. 8 (left) compares the running time of our algorithm (25 buckets per channel) and one it-

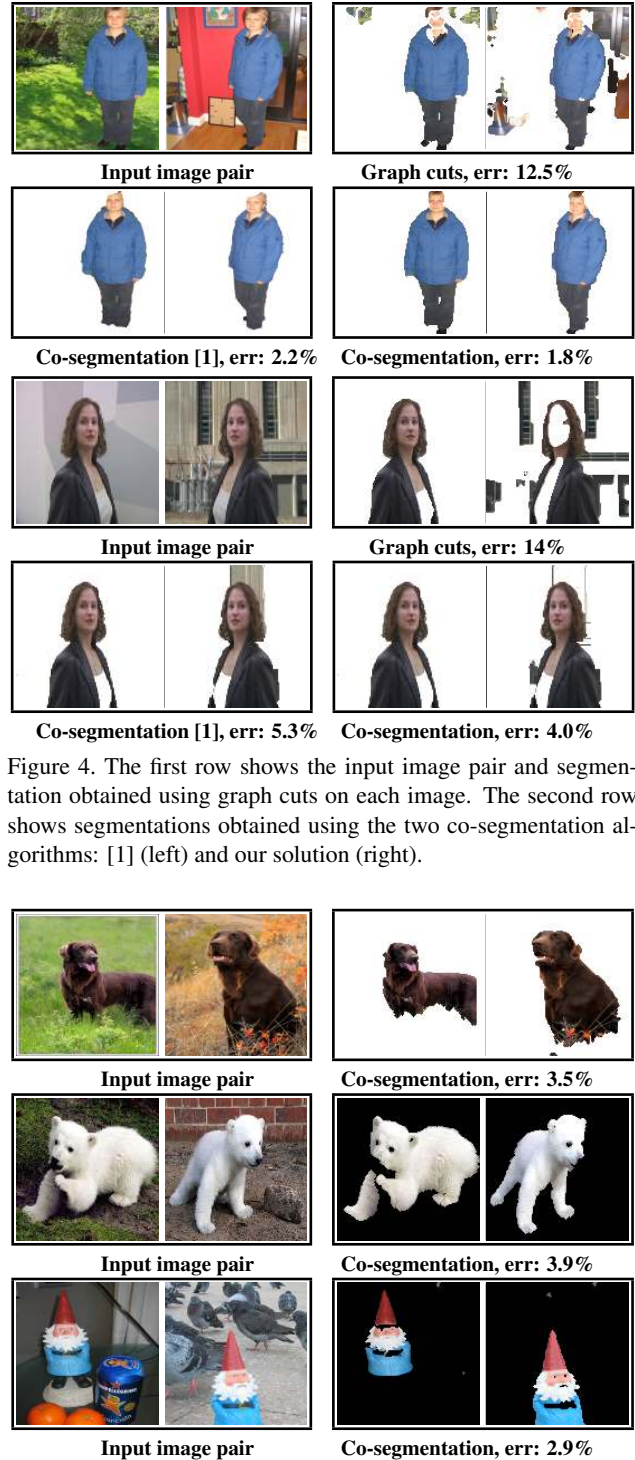


Figure 4. The first row shows the input image pair and segmentation obtained using graph cuts on each image. The second row shows segmentations obtained using the two co-segmentation algorithms: [1] (left) and our solution (right).

Figure 5. Co-segmentation results on images obtained from Flickr with moderate variations in illumination and scale.

eration of [1] (error bars indicate standard deviation) as a function of image size, using -O3 flag with g++ compiler on a GNU/Linux workstation. The plot only includes the time for computing the maximum flow (not for setting up



Figure 6. Change in segmentation as a response to variation in the magnitude of introduced bias,  $\lambda$ .

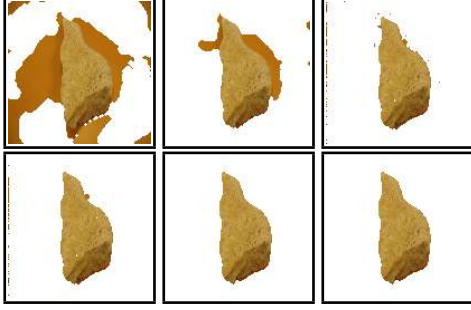


Figure 7. Varying the number of buckets in the histogram from 10 buckets (top-left) to 30 buckets (bottom-right).

the graphs, reading in images). We see that the computational burden for one iteration of both algorithms is almost equivalent. In Fig. 8 (right), we show the running time of the two algorithms (number of iterations of [1] = 5). Since the proposed algorithm requires only one iteration, we see that it offers a noticeable advantage for large image sizes.

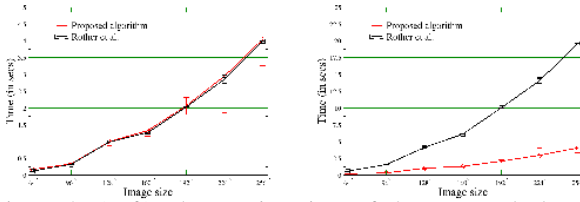


Figure 8. (Left) The running time of the proposed algorithm and one iteration of [1] as a function of the image sizes  $\{64 \times 64, \dots, 256 \times 256\}$ . (Right) The running time of the proposed algorithm and [1] (number of iterations was kept fixed at five).

#### 6.4. Co-segmentation for pathology detection in brain images

Finally, we discuss an application to the extraction of pathologies such as lesions from brain image volumes, important in the study of neurological disorders such as Alzheimer’s disease (AD). Here, the strategy of using a standard segmentation followed by extensive post processing may be too dataset specific, where as a large number of well characterized examples are needed for a supervised learning approach. Also, standard segmentation methods are typically designed to extract the distinct regions from

an image, which makes it difficult to adapt them easily for identifying small pathologies. We observe that the availability of additional similar images (even if unsegmented) can be leveraged to identify such pathologies consistently. In our preliminary experiments, we see that while the size, intensity, texture, and location of lesions (and pathologies) vary between subjects, much of the underlying brain structure remains relatively unchanged from one image to the next if they are affinely registered to a common ‘template’. We wish to extract this similarity from multiple images first (as the foreground), and then view the “residuals” as the pathology (or lesions). Figure 9 shows some preliminary results on four pairs of images. The first and second columns show a 2D slice from the images: while the first image has no apparent pathologies, one or more lesions can be seen in the second image. For these experiments, images with lesions and the set of accompanying images were identified by an expert. In Fig. 9 (third column), we see that co-segmentation gives promising preliminary results where the lesions are accurately identified as the background. While these results are preliminary, they illustrate a new possible application of the algorithm.

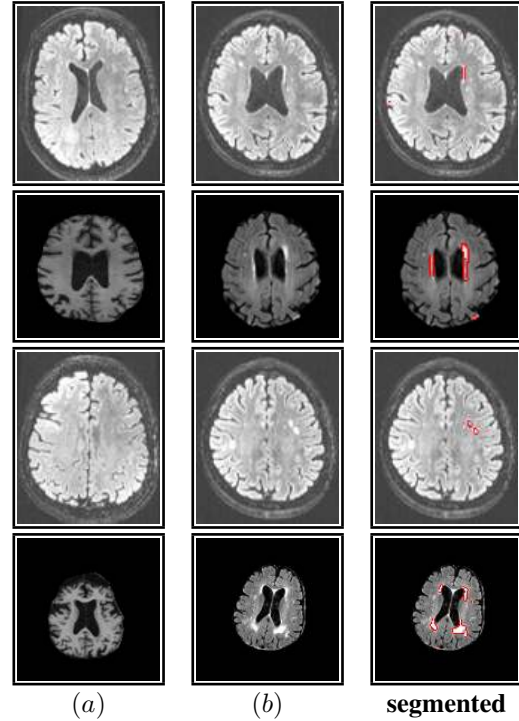


Figure 9. Example segmentation of lesions from brain images where (a) and (b) show the input image pair, and lesions can be seen in the images in (b). The final “background” segmentation is shown in the third column.

## 7. Conclusions

We have proposed an efficient algorithm for the (Co-seg) problem. Our approach is motivated from the carrot or stick philosophy – where rather than penalize the difference (distance) of the two foreground histograms, we reward their similarity (affinity). We demonstrate that the resultant problem has a particularly nice structure, and can be solved optimally in polynomial time using *only one* maximum flow computation. Our solution offers an efficient algorithm for this important problem. On the experimental side, we show that our algorithm performs similarly as existing algorithms, but offers running time savings for larger images. We also show a novel application of co-segmentation to a neuroimaging segmentation problem where it may prove to be valuable.

## Acknowledgments

We gratefully acknowledge the contributions of John Bauml and Lopamudra Mukherjee to this project. Preliminary discussions leading to this paper were conducted at UCLA-IPAM Workshop on Continuous and Discrete Optimization in Computer Vision, we thank IPAM, UCLA, and the organizers of the workshop. This research was supported by NSF award No. DMI-0620677 and DND/DHS award CBET-0736232 (Hochbaum); Wisconsin Comprehensive Memory Program, NIH CTSA award 1UL1RR025011, and UW Graduate School (Singh).

## References

- [1] C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching – incorporating a global constraint into MRFs. In *Proc. of Conf. on Computer Vision and Pattern Recognition*, 2006.
- [2] L. Mukherjee, V. Singh, and C. R. Dyer. Half-integrality based algorithms for cosegmentation of images. In *Proc. of Conf. on Computer Vision and Pattern Recognition*, 2009.
- [3] J. Z. Wang, J. Li, and G. Wiederhold. SIMPLcity: semantics-sensitive integrated matching for picture libraries. *Trans. on Pattern Anal. and Machine Intel.*, 23(9), 2001.
- [4] J. Cui, Q. Yang, F. Wen, Q. Wu, C. Zhang, L. Van Gool, and X. Tang. Transductive object cutout. In *Proc. of Conf. on Computer Vision and Pattern Recognition*, 2008.
- [5] J. Sun, S.B. Kang, Z.B. Xu, X. Tang, and H.Y. Shum. Flash Cut: Foreground Extraction with Flash and No-flash Image Pairs. In *Proc. of Conf. on Computer Vision and Pattern Recognition*, 2008.
- [6] D. S. Cheng and M. A. T. Figueiredo. Cosegmentation for image sequences. In *Proc. of International Conf. on Image Anal. and Processing*, 2007.
- [7] A. C. Gallagher and T. Chen. Clothing cosegmentation for recognizing people. In *Proc. of Conf. on Computer Vision and Pattern Recognition*, 2008.
- [8] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models – their training and applications. *Computer Vision and Image Understanding*, 61(1):38 – 59, 1995.
- [9] A.L. Yuille, P.W. Hallinan, and D.S. Cohen. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8(2):99–111, 1992.
- [10] J. Winn and N. Jojic. Locus: Learning object classes with unsupervised segmentation. In *Proc. of International Conf. on Computer Vision*, 2005.
- [11] L. Cao and L. Fei-Fei. Spatially coherent latent topic model for concurrent object segmentation and classification. In *Proc. of International Conf. on Computer Vision*, 2007.
- [12] O. Chum and A. Zisserman. An exemplar model for learning object classes. In *Proc. of Conf. on Computer Vision and Pattern Recognition*, 2007.
- [13] Y. Schnitman, Y. Caspi, D. Cohen-Or, and D. Lischinski. Inducing Semantic Segmentation from an Example. *Proc. of Asian Conf. on Computer Vision*, 2006.
- [14] P. L. Hammer, P. Hansen, and B. Simeone. Roof duality, complementation and persistency in quadratic 0–1 optimization. *Mathematical Programming*, 28(2):121–155, 1984.
- [15] L. Le Cam and G.L. Yang. *Asymptotics in statistics: some basic concepts*. Springer, 2000.
- [16] D. Comaniciu and P. Meer. Mean shift analysis and applications. In *Proc. of International Conf. on Computer Vision*, 1999.
- [17] D. S. Hochbaum. An efficient algorithm for image segmentation, markov random fields and related problems. *Journal of the ACM*, 48(2):686 – 701, 2001.
- [18] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *Trans. on Pattern Anal. and Machine Intel.*, 23(11):1222–1239, 2001.
- [19] D. S. Hochbaum. Solving integer programs over monotone inequalities in three variables: A framework for half integrality and good approximations. *European Journal of Operational Research*, 2002.
- [20] S. E. Grigorescu, N. Petkov, and P. Kruizinga. Comparison of texture features based on gabor filters. *Trans. on Image Processing*, 2002.
- [21] T. Leung and J. Malik. Recognizing surfaces using three-dimensional textons. In *Proc. of International Conf. on Computer Vision*, 1999.
- [22] V. Kolmogorov, Y. Boykov, and C. Rother. Applications of parametric maxflow in computer vision. In *International Conf. on Computer Vision*, pages 1–8, 2007.
- [23] D. S. Hochbaum. The pseudoflow algorithm: A new algorithm for the maximum-flow problem. *Operations Research*, 56(4):992–1009, 2008.