Half-Integrality based Algorithms for Cosegmentation of Images

Lopamudra Mukherjee¶
mukherjl@uww.edu
Mathematics & Computer Science
Univ. of Wisconsin–Whitewater

Vikas Singh†‡
vsingh@biostat.wisc.edu
Biostatistics & Medical Inform.
Univ. of Wisconsin–Madison

Charles R. Dyer‡†
dyer@cs.wisc.edu
Computer Sciences
Univ. of Wisconsin–Madison

Abstract

We study the cosegmentation problem where the objective is to segment the same object (i.e., region) from a pair of images. The segmentation for each image can be cast using a partitioning/segmentation function with an additional constraint that seeks to make the histograms of the segmented regions (based on intensity and texture features) similar. Using Markov Random Field (MRF) energy terms for the simultaneous segmentation of the images together with histogram consistency requirements using the squared $L_2$ (rather than $L_1$) distance, after linearization and adjustments, yields an optimization model with some interesting combinatorial properties. We discuss these properties which are closely related to certain relaxation strategies recently introduced in computer vision. Finally, we show experimental results of the proposed approach.

1. Introduction

Cosegmentation refers to the simultaneous segmentation of similar regions from two (or more) images. It was recently proposed by Rother et al. [1] in the context of simultaneously segmenting a person or object of interest from an image pair. The idea has since found applications in segmentation of videos [2] and shown to be useful in several other problems as well [3; 4]. The model [1] nicely captures the setting where a pair of images have very little in common except the foreground. Notice how the calculation of image to image distances (based on the entire image) can be misleading in these cases. As an example, consider Fig. 1, where approximately the same object appears in the pair of images. The background and the object’s spatial position in the respective image(s) may be unrelated, and we may want to automatically extract only the coherent regions from the image pair simultaneously. This view of segmentation is also very suitable for biomedical imaging applications where it is important to identify small (and often inconspicuous) pathologies, either for evaluating the progression of disease or for a retrospective group analysis. Here, standard segmentation techniques, which are usually designed to reliably extract the distinct regions of the image, may give unsatisfactory results. However, if multiple images of a particular organ (e.g., brain) are available, the commonality shared across the images may significantly facilitate the task of obtaining a clinically usable segmentation. For instance, since the primary brain structures remain relatively unchanged from one subject to the next, extracting this coherence as the foreground leaves the variation (i.e., patient specific pathology) as the “residuals” in the background.

Figure 1. The same object in different positions in two images with different backgrounds.
fully used for region matching [12]. A generative model was proposed – assuming a Gaussian model on the target histogram (of the to-be-segmented foreground); they suggested maximizing the posterior probability that the foreground models in both images are the same. The formulation results in a challenging optimization problem, and requires an iterative minimization. Some clever observations allow the solution of subproblems using graph-cuts, where it is shown that the objective function value improves (or in the worst case, does not deteriorate) at each iterative step. However, the approximation factor guarantees associated with graph-cuts obtained for each iteration (subproblem) do not carry over to the original optimization. Recently, [13] proposed constructing a Joint Image Graph for soft image segmentation and calculating point correspondences in the image pair. Because the underlying discrete problem is hard, the authors used a relaxation method followed by an iterative two step approach to find the solution.

In this paper, inspired by [1], we also seek to match the histograms1 of the segmented regions. But rather than a generative model (with predefined distribution families) for the histograms, we consider the matching requirement as an algebraic constraint. Our successive formulation includes histogram constraints as additional, appropriately regularized, terms in the segmentation objective function. We analyze the structure of the model using the squared L2 distance (SSD) for measuring histogram similarity, instead of the L1-norm [1]. After linearization and adjustments of the objective function, the constraint matrix of the resultant linear program exhibits some interesting combinatorial properties, especially in terms of the LP solutions of the ‘relaxation’. We observe this by analyzing the determinant of the submatrices of the constraint matrix, which suggests a nice 2-modular structure [14]. The form of the objective function (after linearization) also corroborates the fact that if we choose SSD to specify histogram variations, the problem permits roof-duality relaxation [15; 16] recently introduced in computer vision [17; 18; 19]. Either way, the primary benefit is that unlike the (harder) L1-norm based problem, the LP solution of the new relaxed LP contains only multiples of “half-integral” values. Demonstrating that the cosegmentation problem exhibits these desirable characteristics (without major changes to the underlying objective function) is a primary contribution of this paper. These properties allow a simple rounding scheme that gives good solutions in practice and enables bounding the sub-optimality due to rounding under some conditions. We discuss these aspects in §3.1-§3.3. Finally, we present experimental results on a set of image pairs in §4 and concluding remarks in §5.

1We consider intensity histograms in our implementation, though the proposed model can be directly extended for other types of histograms (such as ones incorporating texture features) as well, as noted in [1].

2. Histogram matching

Let the two input images be \( I_i = [I_i(p)], i \in \{1, 2\}, p \in \{1, \ldots, n\} \), where each image has \( n \) pixels. Let the intensity histogram bins be given by sets \( H_1, \ldots, H_\beta \), where \( H_b \) corresponds to the \( b \)-th bin. For each image \( I_i \), a coefficient matrix \( C_i \) of size \( n \times \beta \) is such that for pixel \( j \) and histogram bin \( H_b \),

\[
C_i(p, b) = \begin{cases} 
1 & \text{if } I_i(p) \in H_b; \\
0 & \text{otherwise}.
\end{cases}
\] (1)

The entry \( C_i(p, b) \) is 1 if pixel intensity \( I_i(p) \) belongs to \( H_b^{(i)} \) (i.e., the intensity of \( I_i(p) \) is in bin \( b \)), where \( i \) refers to the first or the second image. Summing over the columns of \( C_i \) gives the histogram for each image as a vector. Now, consider \( X_1, X_2 \in \mathbb{R}^n \) as a pair of \( \{0, 1\} \) assignment vectors for images \( I_1 \) and \( I_2 \), that specifies the assignment of pixels to foreground and background regions by a segmentation method. For \( i \in \{1, 2\} \), \( X_i(p) = 1 \) if \( I_i(p) \) is classified as foreground and 0 otherwise. We want to cosegment (i.e., by assigning to foreground) two regions from the image pair with the requirement that the two histograms of the foreground pixels are similar. The histogram, \( H^{(i)} \) for image \( I_i \), for the pixels assigned to the foreground is

\[
H_b^{(i)} = \sum_{p=1}^n C_i(p, b) X_i(p) \quad \forall b \in \{1, 2, \ldots, \beta\}, \quad \forall i \in \{1, 2\}.
\] (2)

Since \( X_i(p) \) is 0 if pixel \( p \) is a background pixel, we can simply focus on the histogram of the foreground pixels and penalize the variation. We note that an expression similar to (2) was discussed in a technical report accompanying [1] to obtain a supermodularity proof.

Modeling image segmentation problems as maximum a posteriori (MAP) estimation of Markov Random Fields with pairwise interactions has been very successful [20; 21] and gives good empirical results. It also seems suitable for the segmentation objective here. In this framework, given an image \( I \), each pixel (random variable) \( p \in I \), can take one among a discrete set of intensity labels, \( L = \{L_1, L_2, \ldots, L_k\} \). The pixel must (ideally) be assigned a label \( f_p \in L \) similar to its original intensity, to incur a small deviation (or data) penalty, \( D(p, f_p) \); simultaneously two adjacent (and similar) pixels, \( p, q \), must be assigned similar labels to avoid a high separation (or smoothness) penalty \( W(p, q) : p \sim q \) (where \( \sim \) indicates adjacency in a chosen neighborhood system). This gives the following objective function for arbitrary number of labels:

\[
\min \sum_{p \sim q} W(p, q) Y(p, q) + \sum_{f_p \in L} \sum_{p=1}^n X(p, f_p) D(p, f_p) \quad p, q \in I,
\] (3)

where \( Y(p, q) = 1 \) indicates that \( p \) and \( q \) are assigned to different labels and \( X(\cdot, \cdot) \) gives the pixel-to-label assignments. Proceeding from (2) and allowing for cases where the histograms do not match perfectly, we can specify the
following simple binary model where $X, Y \in \{0, 1\}$ (labels are foreground or background) using an error tolerance, $\epsilon$

$$\min \sum_{p=1}^{2} \sum_{q=1}^{2} W_i(p, q) Y_i(p, q) + \sum_{i=1}^{n} \sum_{p=1}^{n} X_i(p) D_i(p)$$

subject to

$$\sum_{p=1}^{2} \sum_{q=1}^{2} C_i(p, b) X_i(p) - \sum_{p=1}^{n} C_2(p, b) X_2(p) \leq \epsilon \quad \forall b,
$$

$$|X_i(p) - X_i(q)| \leq Y_i(p, q) \quad \forall i, \forall p \sim q.$$  

Here, $D_i(p)$ is a simplified form of the function $D_i(p, f_p)$ for the figure-ground segmentation case. We set $D_i(p) = D_i(p, f_g) - D_i(p, bg)$ where ‘fg’ (and ‘bg’) indicates foreground (and background). The objective also has a constant term given by $\sum_{p, q} D_i(p, bg)^2$ that makes it positive.

### 3. Successive model

The previous model may be modified by including the histogram constraint as an additional (regularized) term in the objective function:

$$\min \sum_{i=1}^{2} \sum_{p=1}^{2} W_i(p, q) Y_i(p, q) + \sum_{i=1}^{n} \sum_{p=1}^{n} X_i(p) D_i(p) + \lambda R(C_1, C_2)$$

where $\lambda$ is the regularizer controlling the relative influence of the histogram difference in the objective and $R(C_1, C_2)$ measures the difference of the two foreground histograms. Including this term (or the constraints in (4)), however, makes the optimization more challenging. Further analysis shows that the difficulty of the resultant optimization model can be attributed not only to the inclusion of the extra penalty in the objective but to the choice of the norm in this additional term. Using the squared $L_2$ distance (rather than $L_1$) has significant advantages as we will discuss shortly. First, using squared $L_2$ distances, we can represent the form of histogram differences as

$$R(C_1, C_2) = \sum_{b=1}^{\beta} \left( \sum_{p=1}^{n} C_1(p, b) X_1(p) - \sum_{q=1}^{n} C_2(q, b) X_2(q) \right)^2$$

This squared term can be linearized as follows. Consider two corresponding bins $H_b^{(1)}$ and $H_b^{(2)}$ in the two images. Let $|H_b^{(1)}| = n_{v_b}$ and $|H_b^{(2)}| = n_{\bar{v}_b}$, and the number of foreground pixels after the segmentation in those bins is $v_b$ and $\bar{v}_b$ respectively. Then, the squared term estimates the sum of $(v_b - \bar{v}_b)^2$ over $b$, i.e., $\sum_b (v_b \bar{v}_b - 2v_b \bar{v}_b + \bar{v}_b \bar{v}_b)$. We may use an auxiliary variable, $Z_i(p, q)$, such that $Z_i(p, q) = 1$ if both nodes $p$ and $q$ belong to bin $H_b^{(1)}$ and are also part of the foreground in image $I_1$, and 0 otherwise. Then, summing over all such $Z_i$, gives the terms $v_b \bar{v}_b$ and $\bar{v}_b \bar{v}_b$. Similarly, we may use another auxiliary variable, $V(p, q)$, which is set to 1 if node $p$ (and node $q$) belongs to bin $H_b^{(1)}$ (and bin $H_b^{(2)}$), and is part of the foreground in image $I_1$ (and image $I_2$). To specify the linearized representation, let us first introduce some notation.

Let $(p, q)_i$ denote those pairs $(p, q)$ that satisfy $C_i(p, b) = 1$ and $C_i(q, b) = 1$ for some bin $H_b^{(1)}$ and image $i$ (i.e., intra-image links). Also, $(p, q)$ denotes those pairs $(p, q)$ that satisfy $C_1(p, b) = 1$ and $C_2(q, b) = 1$ for bins $H_b^{(1)}$ and $H_b^{(2)}$ (i.e., inter-image links). To summarize, this procedure introduces ‘lifting’ variables for linearization (see [22; 17] for other examples) and allows rewriting the model as

$$\min \sum_{i=1}^{2} \sum_{p=1}^{2} W_i(p, q) Y_i(p, q) + \sum_{i=1}^{n} \sum_{p=1}^{n} X_i(p) D_i(p) + \lambda \sum_{b=1}^{\beta} \left( \sum_{i=1}^{2} \sum_{p, q} Z_i(p, q) - 2 \sum_{p, q} V(p, q) \right)$$

subject to

$$X_i(p) - X_i(q) \leq Y_i(p, q), \forall i, \forall p \sim q,
$$

$$X_i(p) + X_i(q) \leq Z_i(p, q) + 1, \forall (p, q),
$$

$$X_i(p) \geq V(p, q), \forall i, \forall p \sim q,
$$

$$\sum_{i=1}^{2} \sum_{p, q} Z_i(p, q) \leq \sum_{i=1}^{2} \sum_{p, q} V(p, q), \forall (p, q),$$

$X, Y, Z, V \in \{0, 1\}$.

### 3.1. Properties of the constraint matrix in (7)

The preferred alternative to solving the $\{0, 1\}$ integer program (IP) in (7) directly (using branch-bound methods) is to relax the $\{0, 1\}$ requirement to $[0, 1]$, and then obtain an integral solution from the $[0, 1]$ solution via rounding. In general, little can be said about the real valued solution. In some special cases, the situation is better and the cosegmentation problem described in (7) belongs to this category.

If the model has only (monotone) constraints of the form in the first set of inequalities in (7), the constraint matrix is totally unimodular (i.e., the determinants of each of its square submatrices are in $\{0, \pm 1\}$). By the Hoffman-Kruskal theorem [23], the optimal vertex solution of the linear program is integral. In cosegmentation, the constraints, $X_i(j) + X_i(l) \leq Z_i(j, l) + 1$, are non-monotone, and spoil unimodularity. Fortunately, the model still retains a desirable structure – first, we derive this by analyzing the modularity properties of the constraint matrix. Later, we discuss how the objective function may also be interpreted to recognize the nice structure of the model. These properties will help show that the values in the optimal LP solution cannot be arbitrary reals in $[0, 1].$ We first provide some definitions.

**Definition 1 (Nonseparable Matrix)** A matrix is nonseparable if there do not exist partitions of the columns and rows to two (or more) subsets $C_1, C_2$ and $R_1, R_2$ such that all nonzero entries in each row and column appear only in the submatrices defined by the sets $R_1 \times C_1$ and $R_2 \times C_2$.

Let the constraint matrix of (7) be denoted as $A$. Below, we outline the key properties of the constraint matrix for the cosegmentation problem.

---

2The data term is $X_i(p) D_i(p, f_g) + (1 - X_i(p)) D_i(p, bg) = D_i(p, bg) + (D_i(p, f_g) - D_i(p, bg)) X_i(p)$. 

---

2030
Lemma 1 A is a nonseparable matrix.

Proof: The entries in X refer to the pixels of an image. For two pixels that are adjacent w.r.t. the chosen neighborhood system \( N \) (e.g., four neighborhood), the columns of their corresponding X entries will be non-zero in at least one row (for the constraint where they appear together). Then, they must be in the same partition (either \( R_1 \) or \( R_2 \)). Other ‘neighbors’ of these pixels must also be in the same partition. Because each pixel is reachable from another via a path in \( N \), the same logic applied repeatedly shows that the X columns (correspond to pixels) must be in the same partition. Each Y and Z variable appears in at least one constraint (non-zero entry in A) together with X, and so must belong in the same partition as X. Each V variable appears in two constraints, with \( X_1(\cdot) \) and \( X_2(\cdot) \), and so must also be in the same partition. Therefore, A is non-separable. □

Theorem 1 The determinant of all submatrices of A belongs to \( \{-2, -1, 0, 1, 2\} \), i.e., its absolute value is bounded by 2.

Proof: We prove by induction. Since the entries in A are drawn from \( \{-1, 0, 1\} \), the 1 \( \times \) 1 matrix case is simple. Now assume it holds for any \( m - 1 \times m - 1 \) submatrix and consider \( m \times m \) submatrices. By construction, A (and any non-singular square submatrix) has at most three non-zero entries in a row. Let \( \bar{A} \) be a \( m \times m \) non-singular submatrix of A. To obtain the result, we must consider the following three cases.

Case 1: \( \bar{A} \) has one non-zero entry in any row/column. 
Case 2: \( \bar{A} \) has two non-zero entries in every row and column. 
Case 3: \( \bar{A} \) has three non-zero entries in any row. 

It is easy to verify that the structure of A rules out other possibilities. For presentation purposes, we will start with Case 3 and consider Case 2 last.

Case 3: \( \bar{A} \) has three non-zero entries in any row. First, observe that the rows with three non-zero entries must come from the first three constraints in (7). In each such constraint, there is at least one variable, \( Y_i(j, l) \), or \( Z_i(j, l) \), that does not occur in any other constraint. Therefore, the corresponding columns of that variable must have only one non-zero entry, say at position \((p, q)\). Let \( M_{u,v} \) denote the matrix obtained from \( M \) by deleting row \( u \) and column \( v \). The determinant of \( \bar{A} \) is expressed as \( A(1,1) \det(A_{1,1}) \), where \( A_{1,1} \) is \( m - 1 \times m - 1 \). Now, \( A(1,1) \in \{\pm 1, 0\} \) and \( \det(A_{1,1}) \in \{\pm 2, \pm 1, 0\} \), so the product of these terms is in \( \{\pm 2, \pm 1, 0\} \).

Case 1: \( \bar{A} \) has 1 non-zero entry in any row. This can be shown using the same argument as in case 3 above.

Case 2: \( \bar{A} \) has 2 non-zero entries in every row and column. This case was proved by Hochbaum et al. [24] (Lemma 6.1) and the same idea can be applied here.

Wlog, we may assume that the two non-zero entries in row \( i \) of \( \bar{A} \) are in columns \( i \) and \((i + 1) \mod m \) (due to Lemma 1). Hence, \( \det(\bar{A}) = A(1,1) \det(A_{1,1}) - (-1)^m A(m,1) \det(A_{m,1}) \). The submatrix determinants equal 1 since they are both triangular matrices (with non-zero diagonal elements). Thus, \( \det(\bar{A}) \in \{\pm 2, \pm 1, 0\} \). □

Corollary 1 The model in (7) has super-optimal half integral solutions, i.e., each variable in the optimal LP solution is in \( \{0, \frac{1}{2}, 1\} \).

Corollary 1 shows that A has 2-modular structure with half-integral solutions [14]. This property leads to a two-approximation for a wide variety of NP-hard problems including vertex cover and many variations of 2-SAT [24]. Notice that if the objective function has only positive terms, we can round all the \( \frac{1}{2} \) variables up to 1, leave the integral variables unchanged, and still ensure that the value of the objective is within a factor of two of the optimal solution [24]. This is not applicable in our case due to the negative term in the objective function. However, we can still obtain approximations if the half integral solution satisfies some conditions, as we show in §3.3.

3.2. Pseudo-Boolean optimization

In the last section, we analyzed the constraint matrix of the LP to derive desired properties. An analogous approach is to analyze the objective function in (5), which is given by the MRF terms (submodular) and the histogram variation (non-submodular). To facilitate this discussion, first recall that a Pseudo-Boolean (PB) function has the form:

\[
\sum_{S \subseteq \mathcal{U}} c_S \prod_{j \in S} x_j
\]

where \( \mathcal{U} = \{1, 2, \ldots, n\} \), \( x = (x_1, x_2, \ldots, x_n) \in \mathbb{B}^n \) denotes a vector of binary variables, \( S \) is a subset of \( \mathcal{U} \), and \( c_S \) denotes the coefficient of \( S \). That is, a function \( f : \mathbb{B}^n \to \mathbb{R} \) is called a pseudo-Boolean function. If the cardinality of \( S \) is upper bounded by 2, the corresponding form is

\[
f(x_1, x_2, \ldots, x_n) = \sum_i c_i x_i + \sum_{i,j} c_{ij} x_i x_j
\]

These are Quadratic Pseudo-Boolean functions (QPB). The histogram term in (6) from our model can also be written in this form. If the objective permits a representation as a QPB, an upper (or lower) bound can be derived using roof (or floor) duality [16], recently utilized in several vision problems [18; 19; 17]. Obtaining a solution then involves representing each variable as a pair of literals, \( x_i \) and \( \tilde{x}_i \), each representing a node in a graph where edges are added based on the coefficients of the terms in the corresponding QPB. A max-flow/min-cut on this new graph yields a part of the optimal solution, i.e., the \( \{0, 1\} \) values in the solution.
(called ‘persistent’) are exactly the same as an optimal solution to the problem instance. The unassigned variables correspond to half-integral values as discussed in the previous section. Independent of the method employed, once such a super-optimal/half-integral solution [24] is found, the challenge is to derive an integral solution via rounding.

### 3.3. Rounding and approximation

The approximation depends critically on how the variables are rounded. If we fix the \{0, 1\} variables, and round \( \frac{1}{2} \)'s to 1, a good approximation may be obtained. However, such a solution may not be feasible w.r.t. the constraints in a worst case setting (although in practice, a simple rounding heuristic exploiting half-integral solutions may work). We discuss this issue next where we fix the \( X \) variables. Note that “fixing” the set of \{0, 1\} variables (as in [25; 24]) is the same as “persistency” in the pseudo-Boolean optimization literature, i.e., only the \( \frac{1}{2} \) valued variables are modified.

We refer to the block of \( X \) variables in the solution vector as \( X \) for convenience; \( X \) includes \( X_1 \) and \( X_2 \). The corresponding block in the optimal LP solution is given as \( X^*_1 \) and \( X^*_2 \) (in the present context we will refer to \( X \)'s as sets). Clearly, \( X^*_i = X_1^{(i)(0)} \cup X_1^{(i)(1)} \cup X_2^{(i)(\frac{1}{2})} \), where \( X_1^{(i)(0)} \), \( X_1^{(i)(1)} \) and \( X_2^{(i)(\frac{1}{2})} \) refers to the 0, 1 and \( \frac{1}{2} \) entries in \( X^* \) respectively. Let \( X_1^{(i)(\frac{1}{2})} \subseteq X_1^{(i)(\frac{1}{2})} \) refer to those entries that are in histogram bin \( b \) in image \( i \) (for presentation purposes, we assume \( H_b \) in image \( i \) corresponds to \( H_b \) in image 2, and so we drop the image superscript). A subset of the constraints in (7) relating \( X_1^{(i)(\frac{1}{2})} \) and \( X_2^{(i)(\frac{1}{2})} \) are

\[
X_1^*(p) + X_1^*(q) \leq Z_i^*(p, q) + 1, \quad X_1^*(p), X_1^*(q) \in X_1^{(i)(\frac{1}{2})},
\]

\[
X_1^*(p) \geq V^*(q, r), \quad X_1^*(q) \in X_1^{(i)(\frac{1}{2})}, \quad (9)
\]

\[
X_2^*(r) \geq V^*(q, r), \quad X_2^*(r) \in X_2^{(i)(\frac{1}{2})}, \quad (10)
\]

\[
X_1^*(p) \geq V^*(q, r), \quad X_1^*(p) \in X_1^{(i)(\frac{1}{2})}, \quad (11)
\]

\[
X_2^*(r) \geq V^*(q, r), \quad X_2^*(r) \in X_2^{(i)(\frac{1}{2})}. \quad (12)
\]

First, consider the \( \frac{1}{2} \)-valued entries. Since \( X_1^*(p), X_1^*(q) \) and \( X_2^*(r) \) are all \( \frac{1}{2} \), and (7) is a minimization, \( Z_i^*(p, q) \) is 0 and \( V^*(q, r) \) is \( \frac{1}{2} \) at optimality. Setting \( X_1^*(p), X_1^*(q) \) and \( X_2^*(r) \) (to 1) satisfies (9)-(12), regardless of the value of \( V^*(\cdot, \cdot) \), but (8) is violated if \( Z_i^*(p, q) \) is unchanged. On the other hand, if we round \( X_1^*(q) \) (or \( X_1^*(p) \)) to 0, (8) is satisfied, but (9) or (11) is violated. Therefore, to ensure a feasible solution, we must round a few \( \frac{1}{2} \) variables \( (V \) and \( X \) to 0, and set a few \( Z \) variables (which were 0 in the optimal solution) to 1. In the general case, this might increase the objective function arbitrarily. However, under some conditions, we can still bound the gap.

For convenience, let \( a_i = |X_1^{(i)(1)}|, d_i = |X_1^{(i)(\frac{1}{2})}| \) and \( c_i = |X_1^{(i)(0)}| \). In the rounding, we do not change the \( \{0, 1\} \) \( X^* \) variables; only the half-integral variables are rounded to 1 or 0. Assume that a rounding scheme, \( R \), sets \( d_1^{(1)} \) entries to 1 and \( d_1^{(0)} \) entries to 0, so \( d_i = d_1^{(1)} + d_1^{(0)} \). Consider the histogram mismatch penalty in the objective function. In the optimal solution, let this term be \( O_{H_b} = \lambda(O_Z - O_V) = \lambda \left( \sum_{i=1}^{2} \sum_{(p,q)} Z_i^*(p, q) - 2 \sum_{(p,q)} V_{pq}^* \right) \) for \( H_b \). In terms of \( a_i, d_i \) and \( c_i \),

\[
O_Z = \sum_{i=1}^{2} \sum_{(p,q) \in H_b} Z_i^*(p, q) = \sum_{i=1}^{2} (a_i^2 + \frac{1}{2} a_i c_i);
\]

\[
O_V = \sum_{(p,q) \in H_b} V_{pq}^* = a_1 a_2 + \frac{1}{2} a_1 d_1 + \frac{1}{2} a_2 d_1
\]

Let the increase in \( O_{H_b} \) due to rounding be \( \rho = \rho_Z + \rho_V \), where \( \rho_Z \) (and \( \rho_V \)) is the increase due to \( Z \) (and \( V \)). These can be expressed as

\[
\rho_Z = \sum_{i=1}^{2} \left( (d_1^{(1)} - d_1^{(0)})^2 - \frac{1}{2} a_i (d_1^{(0)} - \frac{1}{2}) \right);
\]

\[
\rho_V = \frac{1}{2} (d_1 a_2 + d_2 a_2 + d_1 d_1 - 2d_1 a_2 + 2d_1 a_2 - 2a_2 d_1 - 2a_2 d_1)
\]

For \( i \in \{1, 2\} \), we can prove the following result:

**Lemma 2** If \( a_1 \geq \alpha a_2, a_1 \geq \alpha d_1, d_1 \geq d_2, \) then \( \rho_Z + 2\rho_V \leq \xi \cdot (O_Z^* - 2O_V^*) \) where \( \xi = \max \left( \frac{3(\alpha+1)}{2(\alpha^2-2\alpha-1)}, \frac{4}{\alpha-2} \right) \). For \( \alpha = 3, \rho_Z + 2\rho_V \leq 4 \cdot (O_Z^* - 2O_V^*) \).
4. Experimental results

We performed evaluations of our cosegmentation algorithm on several image pairs used in [1] (see http://research.microsoft.com/~carrot/software.htm) together with additional image pairs that we collected from miscellaneous sources. The complete data will be made publicly available after publication. The image sizes were 128 × 128, and we used the RBF kernel to calculate the data $D(\cdot)$ and smoothness costs $W(\cdot, \cdot)$ in (7). In the current implementation, histogram consistency was enforced using RGB intensities and gradients only, but this can be extended to use texture features. Notice that the size of our problem is dependent on the number of bins used. Using few bins makes the problem size large (as a function of the number of pairs in each bin). Also, such a histogram is insufficient to characterize the image properly: dissimilar pixels may now belong to the same bin. This makes histogram constraints counterproductive. On the other hand, too many bins do not enforce the consistency properties strongly. We found using 30-50 bins per color channel for our experiments achieves a nice balance between these competing effects, and solving the LP on a modern workstation using CPLEX takes 15-20s. In this section, we cover (a) qualitative results of segmentation, (b) error rate, i.e., percentage of misclassified pixels (using hand-segmented images as ground truth), and (c) empirical calculation of the loss in optimality due to rounding. For (a) and (b), we also illustrate results from a graph-cuts based MRF segmentation method (GC) [20] applied on the images independently as well as results from the cosegmentation algorithm of [1].

For comparison with [1], we used the trust region graph cuts approach (TRGC) using an implementation shared with us by the authors of [1]. In TRGC, the energy is iteratively minimized by expressing the new configuration of the non-submodular part of the function as a combination of an initial configuration of the variables and the solution from the previous step. Therefore, it requires an initial segmentation of the images. For this purpose, we used the solution of graph cuts based MRF segmentation (without histogram constraints). Also, since it is difficult to optimize both images simultaneously using the $L_1$ norm, the approach in [1] keeps one image fixed and then optimizes the second image w.r.t. the first, and repeats this process. In our experiments, we found that either (a) the solution converged in under five iterations or (b) in some cases a poor segmentation generated in one of the early iterations adversely affected the subsequent iterations. To avoid the second problem, the number of iterations was kept small. We evaluated various values of the regularizer $\lambda$, and report the settings that correspond to the best results.

In Fig. 2, we show cosegmentation results of the three approaches for a set of image pairs with a similar object in the foreground but different backgrounds. The first two columns correspond to the MRF solution (without histogram constraints), columns three and four illustrate solutions obtained using our approach, and the last two columns show the segmentations from [1]. In the first row (Stone), the stone in image-2 is larger (which gives dissimilar foreground histograms), but the algorithm successfully segments the object in both images. Notice that our method compares well to [1] for this image pair. For the next image pair (Banana), our algorithm is able to recognize the object in both images but shows a significant improvement in image-2 compared to GC with the same parameter settings and is comparable to the results of TRGC. For the third image pair (Woman), our algorithm segments roughly the same foreground as GC for image-1. For image-2, several regions in the background with similar intensities as the foreground are incorrectly labeled by GC. Using the histogram constraints eliminates these regions from the segmentation yielding a cleaner and more accurate segmentation. On this image pair, our solution is better than TRGC. In the fourth (Horse) and fifth (Lasso) image pairs, GC does not perform well on image-1, but yields similar results as our method on image-2. Note that the object (foreground) in Lasso image-1 is difficult to discern from the background making it a challenging image to segment. However, using the histogram constraints, we are able to obtain a reasonable final segmentation. When compared to TRGC, our segmentation seems to correctly identify more foreground pixels for Lasso image-1. Table 1 shows the error rate (i.e., the average pixel misclassification error for each image pair). These were generated by comparing the segmentation with hand segmented images. In general, the misclassification error rate was $1 - 5\%$ as shown in Table 1.

Values for $\lambda$. The specific value of $\lambda$ used in our experiments was determined empirically by iterating over four possible values: starting from $\lambda = 10^{-4}$ and increasing it by an order of magnitude until $\lambda = 1.0$. The extracted foregrounds were then compared using mutual information, and $\lambda = 0.01$ was found to work best for our experiments. We show a plot of this behavior in Fig. 4. As the images suggest, $\lambda$ values at either end of the range $[10^3, 10^{-4}]$ are not ideal for cosegmentation. When $\lambda$ is too large ($\geq 1.0$), the foreground becomes smaller and less smooth. This is because the histogram term overwhelms the MRF terms and returns only those foreground pixels where the histogram bins match perfectly. We found that if $\lambda \leq 10^{-4}$, the results are similar to those obtained by graph-cuts segmentation. Also, Fig. 4 shows that the segmentations are relatively stable for wide range of $\lambda$ values. This makes it relatively easy to select a reasonable $\lambda$ value using our approach. We note that in our experiments, cosegmentation results using the $L_1$ norm seem to be more sensitive to the value of $\lambda$.

Half-integrality/inconsistent pixels. In the proposed
Figure 2. The images in columns (a) and (b) are solutions from independent graph-cuts based segmentations on both images, the pair of images in columns (c) and (d) are solutions from our cosegmentation algorithm applied on the images simultaneously, and images in (e) and (f) are solutions obtained from the algorithm in [1]. The segmentation is shown in blue.

Table 1. Misclassification errors (% of pixels misclassified) in the segmentation and empirical approximation estimates of the solution.

<table>
<thead>
<tr>
<th>Instance</th>
<th>our error rate ($\lambda = 0.01$)</th>
<th>approximation</th>
<th>GC error rate</th>
<th>error rate [1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stone</td>
<td>1.56%</td>
<td>1.04</td>
<td>3.57%</td>
<td>1.92%</td>
</tr>
<tr>
<td>Banana</td>
<td>3.02%</td>
<td>1.02</td>
<td>7.75%</td>
<td>3.33%</td>
</tr>
<tr>
<td>Woman</td>
<td>2.14%</td>
<td>1.06</td>
<td>&gt; 10%</td>
<td>≈ 10%</td>
</tr>
<tr>
<td>Horse</td>
<td>4.80%</td>
<td>1.02</td>
<td>5.70%</td>
<td>4.92%</td>
</tr>
<tr>
<td>Lasso</td>
<td>2.87%</td>
<td>1.03</td>
<td>7.9%</td>
<td>3.72%</td>
</tr>
</tbody>
</table>

Figure 3. The extracted foreground (after MI-based affine registration) for the pair of stone images as a function of $\lambda$.

5. Conclusions

We propose a new algorithm for the cosegmentation problem. The model uses an objective function with MRF terms together with a penalty on the sum of squared differences of the foreground regions’ histograms. We show that if SSD is used as the penalty function for histogram mismatch, the optimal LP solution is comprised of only $\{0, \frac{1}{2}, 1\}$ values. Half integrality leads to a simple round-
ing strategy that gives good segmentations in practice and also allows an approximation analysis under some conditions. The proposed approach also permits general appearance models and requires no initialization.

Acknowledgments

The authors are grateful to Dorit S. Hochbaum and Sterling C. Johnson for a number of suggestions and improvements. V. Singh was supported in part through a UW-ICTR CTSA grant, and by the Wisconsin Comprehensive Memory Program. C. R. Dyer was supported in part by NSF grant IIS-0711887.

References


