On Multiple Foreground Cosegmentation

Gunhee Kim Eric P. Xing
School of Computer Science, Carnegie Mellon University
{gunhee,epxing}@cs.cmu.edu

Abstract

In this paper, we address a challenging image segmentation problem called multiple foreground cosegmentation (MFC), which concerns a realistic scenario in general Web-user photo sets where a finite number of K foregrounds of interest repeatedly occur over the entire photo set, but only an unknown subset of them is presented in each image. This contrasts the classical cosegmentation problem dealt with by most existing algorithms, which assume a much simpler but less realistic setting where the same set of foregrounds recurs in every image. We propose a novel optimization method for MFC, which makes no assumption on foreground configurations and does not suffer from the aforementioned limitation, while still leverages all the benefits of having co-occurring or (partially) recurring contents across images. Our method builds on an iterative scheme that alternates between a foreground modeling module and a region assignment module, both highly efficient and scalable. In particular, our approach is flexible enough to integrate any advanced region classifiers for foreground modeling, and our region assignment employs a combinatorial auction framework that enjoys several intuitively good properties such as optimality guarantee and linear complexity. We show the superior performance of our method in both segmentation quality and scalability in comparison with other state-of-the-art techniques on a newly introduced FlickrMFC dataset and the standard ImageNet dataset.

1. Introduction

With the availability of large amount of online images, often with overlapping contents, it is intuitively more desirable to segment multiple images jointly instead of segmenting each image independently to leverage the enhanced foreground signals due to the co-occurrence of objects in these images. This new approach is known as *cosegmentation*, and has been actively studied in the recent computer vision literature [1, 8, 9, 11, 16, 22, 23] [1].

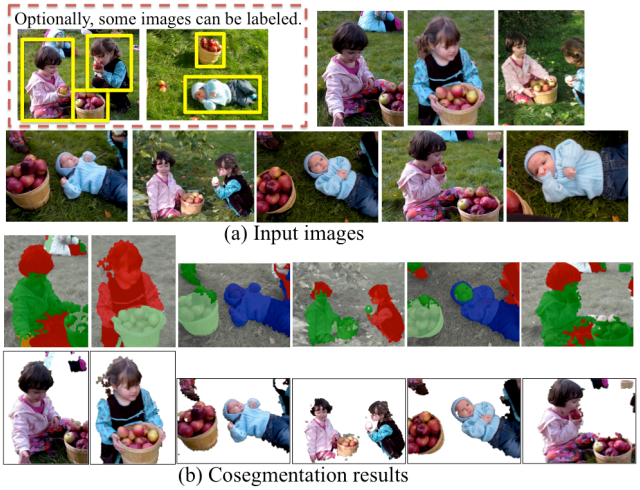

Figure 1. Motivation for multiple foreground cosegmentation. (a) Input images are 20 photos of an *apple+picking* photostream of Flickr. Two girls, one baby, and an apple bucket repeatedly occur in the images, but only a subset of them is shown in each image. (b) The first row shows the color-coded cosegmentation output in which the same colored regions are identified as the same foreground. The second row shows the segmented foregrounds.

However, existing cosegmentation methods still suffer from some limitations in order to be applied to the photo sets of general users. The arguably most limiting one is that every input image would need to contain all the foregrounds for the cosegmentation algorithms to be applicable. Fig.1 shows a typical example that violates this condition. This is an *apple+picking* photostream downloaded from Flickr, and it follows an ordinary photo-taking pattern of a general photographer: a series of pictures about a specific event are taken; the number of objects in a photostream is finite, but they do not appear in every single image. For example, in Fig.1, two girls, one baby, and an apple bucket repeatedly appear in the photostream, but each image includes only an unknown subset of them. Such a *content-misaligned* set of

[1] As to be formalized shortly, the goal of cosegmentation is to divide each of multiple images into non-overlapping regions of *foreground* and *background*. Empirically, the foreground is defined as the common regions that repeatedly occur across the input images [16]. In an interactive or supervised setting [1], the foregrounds are explicitly assigned by a user as the regions of interest, often corresponding to well-defined objects; whereas the background refers to the compliment of foregrounds.

images would not be correctly addressed by existing cosegmentation algorithms. The objective functions in most existing methods were built on the assumption that all input images contain the same objects, without explicitly considering the cases where foregrounds irregularly occur across the images. In order to apply a traditional cosegmentation method to such a photo set, a user is required to first divide her photostream into several groups so that each group contains only photos that have the same foregrounds. This manual preprocessing can be cumbersome, especially when the number of photos is very large (*e.g*. hundreds or more).

In this paper, we propose a combinatorial optimization method, MFC, for cosegmentation that does not suffer from the aforementioned restriction. It allows irregularly occurring multiple foregrounds with varying contents to be present in the image collection, and directly cosegment them. More precisely, we consider the following task:

Definition 1 (Multiple Foreground Cosegmentation). *The multiple foreground cosegmentation (MFC) refers to the task of jointly segmenting K different foregrounds $\mathcal{F}=\{\mathcal{F}^1, \cdots, \mathcal{F}^K\}$ from M input images, each of which contains a different unknown subset of K foregrounds.*

Given the number of foregrounds K and an input image set, our approach automatically finds the most frequently occurring K foregrounds across the image set. Optionally, a user may select the example foregrounds of interest in a couple of images in the form of bounding boxes or pixel-wise annotations. Subsequently, our algorithm segments out every instance of K foregrounds in the input image set.

More specifically, our approach is based on an iterative optimization procedure that alternates between two subtasks: *foreground modeling*, and *region assignment*. Given an initialization for the regions of K foregrounds, the foreground modeling step learns the appearance models of K foregrounds and the background, which can be accomplished by using any existing advanced region classifiers or their combinations. During the region assignment step, we allocate the regions of each image to one of K foregrounds or the background. This is done via a combinatorial auction style optimization algorithm; every foreground and the background bid the regions along with their values of how much the regions are relevant to them. These values are computed by the learned foreground models. Finally, an optimal solution (*i.e*. the allocation of the regions that maximizes the overall value) is achieved in $O(MK)$ time, by leveraging the fact that the candidate regions bidden by foregrounds and the final region assignment can be represented by subtrees of a connectivity graph of regions in the image space. Iteratively, after the region assignment, each foreground model is updated by learning from the newly assigned segments (i.e., regions) to the foreground.

The concept of such an iterative segmentation scheme has been used in some previous work such as [10] and [15]. But the allowance of arbitrary classifiers and their combinations to be plugged in during foreground modeling, and the use of a linear-time algorithm motivated by combinatorial auction for region assignment make our method unique and far more efficient and flexible than earlier ones.

Methods	M	K+1	MFC	Hetero-FG
Ours (MFC)	$\geq 10^3$	Any	O	O
SO [11]	$\geq 10^3$	Any	X	X
UGC [16, 22]	2	2	X	O
SGC [1, 14, 8]	≤ 30	2	X	O
DC [9]	≤ 30	2	X	O

Table 1. Comparison of our algorithm with previous cosegmentation methods. M and K denote the number of images and foregrounds, respectively. *MFC* indicates whether an algorithm is designed to solve the MFC problem in Definition 1. *Hetero-FG* means whether an algorithm can identify a heterogeneous object (*e.g*. a person) as a single foreground. (SO: submodular optimization, UGC: Graph-cuts (unsupervised), SGC: Graph-cuts (supervised), DC: Discriminative clustering).

We test our method on a newly created benchmark dataset, FlickrMFC, with pixel-level ground truth. Each group consists of photos from a Flickr photostream taken by a single user, and contains a finite number of subjects that irregularly appear across the images. Our experiments in Section 4 show that our approach successfully solves the multiple foreground cosegmentation in a scalable way. Moreover, the cosegmentation accuracies are compelling over the state-of-the-art techniques [9, 11, 18] on our novel FlickrMFC dataset and the standard ImageNet dataset [6].

1.1. Relations to Previous work

Cosegmentation: Table 1 summarizes the comparison of our work with previous cosegmentation methods. Our approach has several important features that are beneficial for the cosegmentation of general users' photo sets. Our algorithm is able to handle a large M for scalability and an arbitrary K for highly variable contents of user images. This advantage is also shared with CoSand [11], but our key differences to [11] are as follows. First, the CoSand is a bottom-up approach that relies on only low-level color and texture features, whereas our technique can be merged with any region classification algorithms. Second, the CoSand cannot model a heterogeneous object that consists of multiple distinctive regions (*e.g*. a person) as a single foreground. It can be a limitation to be used for consumer photos because they are likely to contain persons as subjects, which are often required to be segmented as a single foreground. However, our approach does not suffer from these issues.

Our approach can correctly account for multiple foreground cosegmentation in Definition 1, which has not been explicitly addressed by the optimization methods of most previous work [1, 8, 9, 11, 14, 16, 22], as shown in Table 1.

Combinatorial Optimization in Object Detection: Recently, combinatorial optimization techniques have been popularly used in object detection research. Some notable examples include branch-and-bound schemes for efficient subwindow search [12], a Steiner tree based selection of object candidate regions [17], and the maximum-weight connected subgraph for the detection of non-boxy objects [24].

The main purpose of these methods is to efficiently enumerate candidate regions to which object classifiers are applied, which is substantially different from our goal. Consequently, our MFC has a different objective function to be optimized by a different technique, which is based on welfare maximization in combinatorial auction [5].

2. Problem Formulation

Denote the set of input images by $\mathcal{I} = \{I_1, \cdots, I_M\}$. According to Definition 1, we are interested in segmenting out K different foregrounds $\mathcal{F}=\{\mathcal{F}^1, \cdots, \mathcal{F}^K\}$ from all images in $\mathcal{I}$, each with an unknown subset of $\mathcal{F}$. Our algorithm deals with two different scenarios. In the *unsupervised* scenario, a user solely inputs the number K, and our algorithm automatically infers K distinctive foregrounds that are most dominant in $\mathcal{I}$. In the *supervised* scenario, a user can provide bounding-box or pixel-wise annotations for K foregrounds of interest in some selected images.

In our approach, we break the MFC problem defined above into two subproblems, which we solve iteratively: *foreground modeling* and *region assignment*. Foreground modeling learns the appearance models of K foregrounds, and region assignment allocates the regions of each image to one of K foregrounds or the background. Intuitively, given a solution to one of the two subproblems, the other can be easily solved. From an initial region assignment, one can learn K foreground models, which in turn improve region assignment in every image. These two processes alternate until achieving a converging solution.

2.1. Foreground Models

Without loss of generality, we define the k-th foreground (or the background) model as a parametric function $v^k : \mathcal{S} \rightarrow \mathbb{R}$ that maps any region $S \in \mathcal{S}$ in an image to its fitness value to the k-th foreground (*i.e.* how closely the region is relevant to the k-the foreground). If an image I_i is oversegmented as $\mathcal{S}_i$, then $v^k : 2^{|\mathcal{S}_i|} \rightarrow \mathbb{R}$ takes any subset $S \subset \mathcal{S}_i$ as input and returns its value to the k-th foreground. During the region assignment, each foreground model is used to assess how fit a region (or a set of regions) to a foreground, as shown in Fig.2.(a). During the foreground modeling, each foreground model is updated by learning from the segments allocated to the foreground, as shown in Fig.2.(b).

One important objective of our approach is to enable adaptability to any choice or combination of foreground models as plug-ins. Any classifiers or ranking algorithms

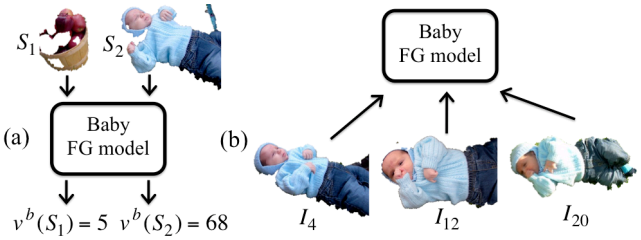

Figure 2. An example of the *baby* foreground (FG) model. (a) A FG model is a parametric function that maps any region to a value to the foreground. (b) After the region assignment, the FG model is updated by learning from the segments assigned to the FG.

can be used as foreground model so long as it can evaluate a region and be updated by learning from the assigned regions. (If we view the foreground model as a classifier, the former is a testing step and the latter is a training step). In this paper, we use two different foreground models - the Gaussian mixture model (GMM) (*i.e.* Boykov-Jolly model [2, 15]) and spatial pyramid matching (SPM) with linear SVM [13]. The former has been a popular appearance model in cosegmentation [1, 22], and the latter is one of baselines for object classification and detection. Table 2 summarizes the region descriptors, model parameters, learning methods, and region valuation of the two foreground models. For both GMM and SPM models, we follow the algorithms proposed in the original papers [2, 15] and [13]. In experiments, the final region score is computed by $v^k(S) = \alpha \cdot v^k_{GMM}(S) + (1-\alpha) \cdot v^k_{SPM}(S)$ by changing α from 0 to 1. Note that thanks to our flexible definition of the foreground model, the simple SPM model can be replaced by the state-of-the-arts deformable part models [7] for better performance.

2.2. Region Assignment

Given the foreground models, the region assignment is performed on individual images separately. The goal of this step is to divide $\mathcal{S}_i$ (*i.e.* the segment set of each image I_i) into disjoint subsets of foregrounds $\mathcal{F}^k_i$ ($k = \{1, \cdots, K\}$) and background (For notational simplicity, we use $\mathcal{F}^{K+1}_i$ for background). Since all foregrounds do not appear in every image, some foregrounds ($\mathcal{F}^k_i$) are empty sets.

Naively, we may distribute each segment $s \in \mathcal{S}_i$ to one of $\mathcal{F}^k_i$ that has the maximum value $v^k(s)$ for it. However, in image segmentation, the value of a segment bundle (*i.e.* a subset of $\mathcal{S}_i$) can be worth more than or less than the sum of values of individual segments. For example, suppose that a black patch is the most valuable to the *cow* foreground. But, if the black patch is combined with a skin-colored patch, this bundle would be more valuable to the *person* foreground than to the *cow* foreground.

Consequently, the region assignment reduces to finding a disjoint partition $\mathcal{S}_i = \bigcup_{k=1}^{K+1} \mathcal{F}^k_i$ with $\mathcal{F}^k_i \cap \mathcal{F}^l_i = \emptyset$ if $k \neq l$, to maximize $\sum_{k=1}^{K+1} v_k(\mathcal{F}^k_i)$. More formally, it corresponds

	GMM	SPM
Region features	A set of RGB colors extracted at every pixel of region S.	A spatial pyramid $h(S)$ (2 levels, 200 visual words of gray/HSV SIFT). The minimum rectangle enclosing S is used as the based pyramid.
Model and learning	A Gaussian mixture with C components. The parameters $\theta^k = \{\pi_c^k, \mu_c^k, \sigma_c^k\}_{c=1}^C$ which are the prior probability, mean, and covariance. The standard EM is used for learning.	A linear SVM is learned by using $\mathcal{F}^k$ as positive data and randomly chosen regions from other foregrounds or background as negative data.
$v^k(S)$	The mean log-likelihood of the RGB descriptors of S to the k-th learned GMM model.	$v^k(S) = \sum_{t=1}^T y_t \alpha_t K(h(S), h(t))$ where $h(t)$ is the histogram of t training region, $y_t \in \{+1, -1\}$ is positive/negative labels, $K(\cdot,\cdot)$ is the histogram intersection kernel, and T is the number of training data.

Table 2. Description of two foreground models – GMM and SPM models.

to the integer program (IL) problem below:

$$\max \sum_{k=1}^{K+1} \sum_{S \subseteq \mathcal{S}_i} v^k(S) x^k(S) \quad (1)$$

$$\text{s.t.} \sum_{k=1}^{K+1} \sum_{s \in S, S \subseteq \mathcal{S}_i} x^k(S) \leq 1, \quad \forall s \in \mathcal{S}_i,$$

$$x^k(S) \in \{0, 1\}$$

where variables $x^k(S)$ describe the allocation of bundle S to k-th foreground $\mathcal{F}_i^k$. (*i.e.* $x^k(S) = 1$ if and only if the k-th foreground takes the bundle S). The first constraint checks whether the assignment is feasible; any segment $s \in S_i$ cannot be assigned more than once.

The region assignment in Eq.(1) requires to check all possible subset $S \subseteq \mathcal{S}_i$. Unfortunately, there are $2^{|\mathcal{S}_i|}$ possible subsets, so enumerating them is infeasible. It is proven in [5] that Eq.(1) is identical to the weighted set packing problem, and thus it is NP-complete and inapproximable.

3. Tractable MFC

In this section, we propose a tractable MFC method that iteratively solves the two subproblems defined in the previous section. The foreground modeling is straightforward, but the region assignment is intractable. Hence, we here focus on developing a polynomial time algorithm to solve the region assignment by taking advantage of structural properties that are commonly observed in the image space.

3.1. Tree-Constrained Region Assignment

Given the $K+1$ foreground models, the region assignment module progresses as follows. First, each image I_i is oversegmented as $\mathcal{S}_i$ as shown in Fig.3.(b). Any segmentation algorithm can be used, and we apply the submodular image segmentation [11] to each image. Given the segment set $\mathcal{S}_i$ of image I_i, each foreground in $\mathcal{F}$ creates a set of *foreground candidates* $\mathcal{B}_i^k = \{B_1^k, \cdots, B_n^k\}$, where every candidate is a tuple $B_j^k = \langle k_j, C_j, w_j \rangle$, where k_j is the index of the foreground that submits candidate j, $C_j \subseteq \mathcal{S}_i$ is a bundle of segments and w_j is its value $w_j = v^k(C_j)$ (See an example in Fig.3.(d)). In this step, we allow each foreground to submit as many candidates as it is willing to take (Section 3.2). Finally, solving the region assignment

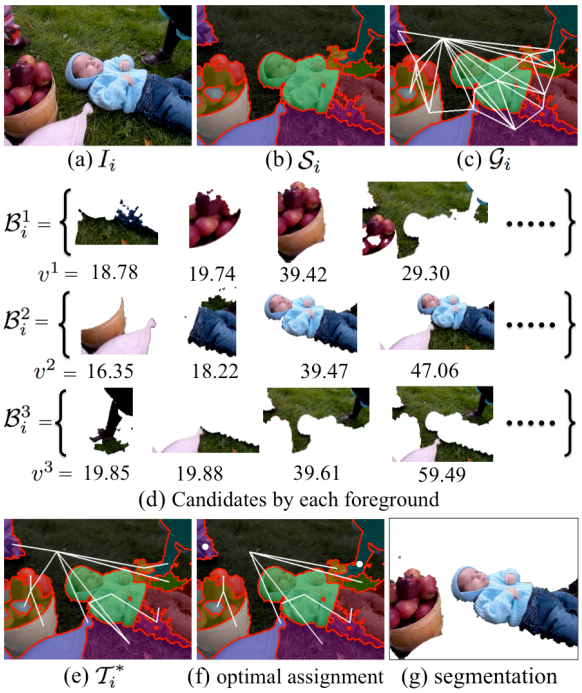

Figure 3. An example of region assignment with *apple bucket* and *baby* foregrounds (FG) and background (BG). (a) An input image I_i. (b) Segment set $\mathcal{S}_i$. (c) Adjacency graph $\mathcal{G}_i$. (d) The set of FG candidates $\mathcal{B}_i$ that are submitted by two FGs and BG. Each candidate is a subtree of $\mathcal{G}_i$, associated with its value. (e) The most likely tree $\mathcal{T}_i^*$ given $\mathcal{B}_i$. (f) The optimal assignment is a forest of subtrees in $\mathcal{B}_i$. (g) The segmentation of two FGs.

in Eq.(1) corresponds to choosing some feasible foreground candidates among all submitted $\mathcal{B}_i = \{\mathcal{B}_i^1, \cdots, \mathcal{B}_i^{K+1}\}$ in order to maximize the overall values[2] (Section 3.3).

There are two possible approaches to make the region assignment problem in Eq.(1) tractable: putting a restriction on value function v^k or a restriction on generating foreground candidates $\mathcal{B}_i$. We explore the latter approach (*i.e.* restriction on $\mathcal{B}_i$) because one of our design goals is to enable flexible choice of foreground models. (*e.g.* it is hard to define any regularity constraints on the output scores of the

[2]Our region assignment is closely related to combinatorial auction [5] with following terminological correspondences: Given a set of segments (items) $\mathcal{S}_i$, $K+1$ foreground models (bidders or buyers) submit a set of foreground candidates (package bids) $\mathcal{B}_i$. The region assignment in Eq.(1) is commonly referred to a *Winner determination problem* or a *Welfare problem* in combinatorial auction literature.

SPM model for arbitrary segment bundles). In the following sections, we will discuss how to achieve the tractability.

Assumption: We assume that a foreground instance in an image is represented by a set of *adjacent* segments. A pair of segments is considered as adjacent if its minimum spatial distance in an image is less than or equal to ρ. This is a reasonable assumption because most foregrounds of interest occupy connected regions in an image. Our approach allows multiple instances (*e.g.* several apple buckets in an image), which are regarded as multiple connected regions.

Suppose that we build an adjacency graph $\mathcal{G}_i{=}(\mathcal{S}_i, \mathcal{E}_i)$ where every segment is a vertex and $(s_l, s_m) \in \mathcal{E}_i$ if $\min d(s_l, s_m) \leq \rho$ (*e.g.* $\rho{=}5$ pixels) for all $s_l, s_m{\in}\mathcal{S}_i$ (See an example in Fig.3.(c)). Then, any connected regions in the image can be represented by subtrees of $\mathcal{G}_i$, and thus the final region assignment $\{\mathcal{F}_i^1, \cdots, \mathcal{F}_i^{K+1}\}$ should be a forest (*i.e.* set) of subtrees (See an example in Fig.3.(f)). Consequently, without loss of generality, we restrict any foreground candidate $B_i \in \mathcal{B}_i$ to be a subtree of the $\mathcal{G}_i$, and our goal of region assignment is to select some B_i that are not only feasible but also maximize the objective of Eq.(1).

3.2. Generating Candidate Sets

In this section, we discuss how each foreground generates a set of foreground candidates $\mathcal{B}_i^k$, each of which is a subtree of $\mathcal{G}_i$ (*i.e.* generating candidates in Fig.3.(d) from $\mathcal{G}_i$ in Fig.3.(c)). In this step, each foreground does not care for the winning chances of its proposals by competing the ones submitted by the other foreground models.

Given the adjacency graph $\mathcal{G}_i$, each foreground samples highly valued subtrees as candidates $\mathcal{B}_i^k$ by using beam search with v^k as a heuristic function and a beam width D [19] (*e.g.* $D{=}10$ in our tests). Algorithm 1 summarizes this process. We start with all unit segments $\forall s \in \mathcal{S}_i$ to be added to $\mathcal{B}_i^k$. In every round, we enumerate all subtrees that can be obtained by adding one edge from previous candidates. The beam width D specifies the maximum number of subtrees to be retained at each round. We only keep top D highly valued subtrees as $\mathcal{B}_i^k$ without consuming too much time on poorly valued ones (See step 3 of Algorithm 1). In practice, this beam search selects good and sufficiently many candidates, because each foreground usually occupies only a part of an image. The computation time of this step per foreground is at most $O(D|\mathcal{S}_i|^2)$, and the number of foreground candidates $|\mathcal{B}_i|$ is at most $(D|\mathcal{S}_i|)$.

3.3. Tractable Region Assignment

Given $\mathcal{B}_i$, we are ready to solve Eq.(1) by choosing some feasible candidates among $\mathcal{B}_i$. For a tractable solution, we first introduce a theorem in [20], which is reformulated to be fit to our context as follows.

Theorem 1 ([20]). *Dynamic programming can solve Eq. (1) in $O(|\mathcal{B}_i||\mathcal{S}_i|)$ worst time if every candidate in $\mathcal{B}_i$ can be represented by a connected subgraph of a tree $\mathcal{T}_i^*$.*

Algorithm 1: Build candidates $\mathcal{B}_i^k$ from $\mathcal{G}_i$ by beam search.

Input: (1) Adjacency graph $G_i = (\mathcal{S}_i, \mathcal{E}_i)$. (2) Value function v^k of the k-th foreground model. (3) D: Beam width.
Output: k-th foreground candidates $\mathcal{B}_i^k$.

1: Set the initial open set to be $\mathcal{O}{\leftarrow}\forall s \in \mathcal{S}_i$. $\mathcal{B}_i{\leftarrow}\forall s \in \mathcal{S}_i$.
for $i = 1$ **to** $|\mathcal{S}_i|{-}1$ **do**
  foreach $o \in \mathcal{O}$ **do**
    2: Enumerate all subgraphs $\mathcal{O}_o$ that can be obtained by adding an edge to o. $\mathcal{O} \leftarrow \mathcal{O}_o$ and $\mathcal{O} \leftarrow \mathcal{O}\backslash o$.
  3: Compute values $v_o \leftarrow v^k(o)$ for all $o \in \mathcal{O}$ and remove o from $\mathcal{O}$ if it is not top D highly valued. $\mathcal{B}_i \leftarrow \mathcal{O}$.

Theorem 1 suggests a linear-time algorithm for region assignment, if $\mathcal{B}_i$ can be organized as a tree. In the foreground candidate set $\mathcal{B}_i$, each $B_i \in \mathcal{B}_i$ is a subtree but its aggregation $\mathcal{B}_i$ may not. Therefore, we reject some $\mathcal{B}_i$ that cause cycles but are not highly valued, because the final solution is a forest of candidate subtrees. The pruned $\mathcal{B}_i$ is denoted by $\mathcal{B}_i^*$. Now we discuss how to obtain $\mathcal{T}_i^*$ and $\mathcal{B}_i^*$ from $\mathcal{B}_i$.

Inferring the tree from the candidate set: Given candidate set $\mathcal{B}_i$ (*i.e.* a set of subtrees), our objective here is to infer the most probable tree $\mathcal{T}_i^*$. It can be formulated as the following maximum likelihood estimation (MLE) in a similar way to tree structure learning (*e.g.* Chow-Liu tree [3]):

$$\mathcal{T}_i^* = \underset{\mathcal{T} \in \mathcal{T}(\mathcal{G}_i)}{\operatorname{argmax}} P(\mathcal{B}_i|\mathcal{T}) \tag{2}$$

where $P(\mathcal{B}_i|\mathcal{T})$ is the data likelihood and $\mathcal{T}(\mathcal{G}_i)$ is the set of all possible spanning trees on $\mathcal{G}_i$.

In the supplementary material, we outline a Chow-Liu style algorithm that computes the most likely tree $\mathcal{T}_i^*$ given $\mathcal{B}_i$ in $O(|\mathcal{B}_i||\mathcal{S}_i|^2)$ time. It is also proven that the solution by this algorithm minimizes the values of rejected B_l in $\mathcal{B}_i$ under the constraint of tree structure as follows:

$$\mathcal{T}_i^* = \underset{\mathcal{T} \subset \mathcal{T}(\mathcal{G}_i)}{\operatorname{argmin}} \sum_{B_l \in \mathcal{B}_i, B_l \not\subset \mathcal{T}_i} v(B_l) \tag{3}$$

Once we obtain $\mathcal{T}_i^*$, we retain only the candidates $\mathcal{B}_i^*$ ($\subset \mathcal{B}_i$) that are subgraphs of $\mathcal{T}_i^*$.

Search Algorithm: As stated in Theorem 1, the optimal solution of Eq.(1) given $\mathcal{B}_i^*$ can be efficiently obtained. We implement a dynamic programming based search algorithm by modifying the CABOB algorithm [21]. We present the pseudocode in the supplementary material.

3.4. The MFC Algorithm

The overall algorithm is summarized in Algorithm 2. We repeat the foreground modeling and region assignment, and stop when the objective value of a new region assignment in Eq.(1) stops increasing. Since we consider the foreground model as a black box, it is difficult to analytically understand the convergence property. However, if we use only

Algorithm 2: Multiple foreground cosegmentation

Input: (1) Input image set $\mathcal{I}$. (2) Number of foregrounds (FGs) K. (3) (In supervised case) annotations $\mathcal{A} = \{\mathcal{A}^1, \cdots, \mathcal{A}^K\}$.
Output: Foregrounds $\mathcal{F}_i = \{\mathcal{F}_i^1, \cdots, \mathcal{F}_i^K\}$ for all $I_i \in \mathcal{I}$.

Initialization

foreach $I_i \in \mathcal{I}$ **do**
  1: Oversegment I_i to $\mathcal{S}_i$ and build adjacency graph $\mathcal{G}_i = (\mathcal{S}_i, \mathcal{E}_i)$ where $(s_l, s_m) \in \mathcal{E}_i$ if $\min d(s_l, s_m) \leq \rho$.

if *unsupervised* **then**
  2: Apply diversity ranking of [11] to the similarity graph of $\mathcal{S}=\bigcup_{i=1}^{M} S_i$ to find K regions $\mathcal{A}=\{\mathcal{A}^1, \cdots, \mathcal{A}^K\}$ that are highly repeated in $\mathcal{S}$ and diverse with respect to one another.

3: Set $\mathcal{F} \leftarrow \mathcal{A}$.

Iterative Optimization

/* *Stopping condition.* */
We stop the iteration if a new region assignment does not increase the objective value (*i.e.* $\sum_{i=1}^{M} \sum_{k=1}^{K+1} v^k(\mathcal{F}_i^k)$ from Eq.(1)).

/* *Foreground Modeling (Any methods can be used).* */
foreach $k = 1{:}K$ **do**
  1: Learn GMM and SPM FG models from $\mathcal{F}^k$ (See Table 2).

/* *Region assignment* */
foreach $I_i \in \mathcal{I}$ **do**
  foreach $k = 1{:}+1K$ **do**
    2: Generate FG candidates $\mathcal{B}_i^k$ by Alg.1 as a set of $B_i^k = \langle k_j, C_j, w_j \rangle$, where k_j is the foreground index, $C_j \subseteq \mathcal{S}_i$ is a subtree of $\mathcal{G}_i$, and $w_j = v^k(C_j)$.
  3: Compute the most probable candidate tree $\mathcal{T}_i^*$ and pruned $\mathcal{B}_i^*$ by Eq.(2) from $\mathcal{B}_i = \bigcup_{k=1}^{K+1} \mathcal{B}_i^k$.
  4: Obtain $\mathcal{F}_i$ to solve region assignment in Eq.(1) by using dynamic programming on $\mathcal{B}_i^*$ (in the supplementary material).

the GMM model as our foreground model, the algorithm is guaranteed to converge at least to a local minimum [15].

The initializations for region assignment are different between supervised and unsupervised settings. In the supervised scenario, the initial foreground regions are labelled by users: $\mathcal{A} = \{\mathcal{A}^1, \cdots, \mathcal{A}^K\}$ where $\mathcal{A}^k$ is the regions annotated as the k-th foreground. In the unsupervised setting, we apply the diversity ranking method of [11] to the similarity graph of $\mathcal{S} = \{\mathcal{S}_1, \cdots, \mathcal{S}_M\}$ to discover the most repeated K regions that are diverse with respect to one another. Note that in the unsupervised setting, commonality of the regions is favored. Hence, when we apply the unsupervised cosegmentation to the images like Fig.3, it is unavoidable to detect grass regions as one of K foregrounds because it is dominant across the input images.

4. Experiments

We evaluate the MFC algorithm using the FlickrMFC dataset and the ImageNet [6] dataset. The FlickrMFC and our MFC toolbox in Matlab can be found at our webpage http://www.cs.cmu.edu/∼gunhee.

4.1. FlickrMFC Dataset

Datasets: The FlickrMFC is a fully manually labeled dataset that consists of 14 groups, each of which includes 10∼20 images. Each group is sampled from a Flick photostream and contains a finite number of repeating subjects that are not presented in every image. The details of FlickrMFC dataset are shown in the supplementary material.

Baselines: As baselines, we use one LDA-based unsupervised localization method [18] (LDA) and two cosegmentation algorithms: CoSand [11] (COS) and discriminative clustering method [9] (DC). Since the two cosegmentation methods are not intended to handle irregularly appearing multiple foregrounds, we first manually divide the images into several subgroups so that the images of each subgroup share the same foregrounds. If an image contains multiple foregrounds, it belongs to multiple subgroups. Then, we apply the methods to each subgroup separately to segment out the common foreground. This is an exact scenario where a conventional cosegmentation is applied to the image sets of multiple foregrounds. The (LDA) [18] was not originally developed for cosegmentation, but it can segment multiple object categories without any annotated information. We use the source codes provided by original authors[3].

Results: Our algorithm is applied in both supervised (MFC−S) and unsupervised (MFC−U) settings. In (MFC−S), we randomly choose 20% of input images (*i.e.* 2∼4 images) to obtain annotated labels for the foregrounds of interest. For the unsupervised algorithms, (MFC−U) and (LDA), it is hard to know the best K beforehand. Thus, we run them by changing K from two to eight, and report the best results.

Fig.4 summarizes the segmentation accuracies on the 14 groups of the FlickrMFC dataset. In the figure, the leftmost bar set is the average performance on 14 groups. The accuracy is measured by the intersection-over-union metric ($\frac{GT_i \cap R_i}{GT_i \cup R_i}$), the standard metric of PASCAL challenges. We observed that the performance of our (MFC−U) is slightly worse than (COS) and (DC) by 2∼3%. Note that (COS) and (DC) are applied to the images of each separate subgroup that shares the same foregrounds. It allows the algorithms to know what foregrounds exist in images beforehand, which is a strong supervision. On the other hand, (MFC−U) is a completely unsupervised; it is applied to the entire dataset without splitting. Our supervised (MFC−S) algorithm, even with a very small number of labeled images, significantly outperformed the competitors by more than 11% over the best of baselines (COS).

Fig.6 shows some examples of cosegmentation from six groups of the FlickrMFC dataset. In each set, we show input images, color-coded cosegmentation output, and segmented foregrounds from top to bottom. The same colored regions in the second row are identified as the same foregrounds, and the meanings of the colors are described below each set.

[3]Codes are available at [11]: http://www.cs.cmu.edu/∼gunhee/, [9] : http://www.di.ens.fr/∼joulin/, [18]: http://www.cs.washington.edu/homes/bcr/projects/mult_seg_discovery/.

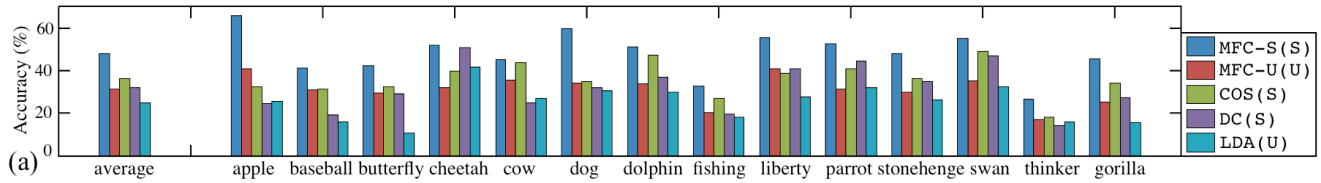

Figure 4. Comparison of segmentation accuracies between our supervised (MFC-S) and unsupervised (MFC-U) approaches and other baselines (COS, DC, LDA) for the FlickrMFC dataset. The S and U indicate whether any annotation information is required (S) or not (U).

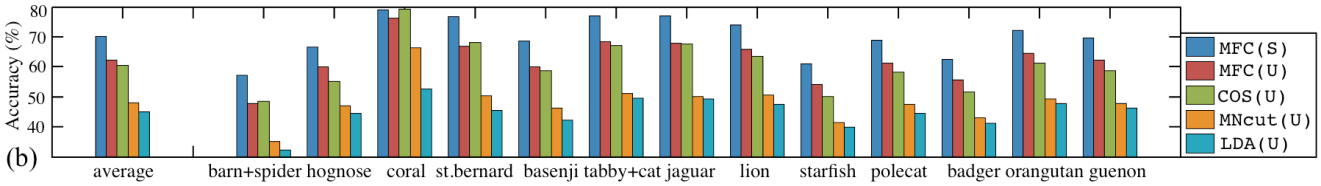

Figure 5. Comparison of segmentation accuracies between our approach and other baselines for the ImageNet dataset.

We made several interesting observations in these examples: First of all, our algorithm correctly treated the multiple foreground cosegmentation in Definition 1. In Fig.6.(a), two girls, a boy, a baby, an apple bucket, pumpkins are intended foregrounds, which are irregularly presented in each image. This is a challenging situation for traditional cosegmentation methods, but our algorithm could successfully segment the foregrounds. As shown in the fourth image of Fig.6.(d), some input images include no foregrounds, which were successfully identified as well. One main source of errors in our experiments was the similarly looking regions; for example, in the first image in Fig.6.(a), the face region of the *girl+red* is allocated to the *girl+blue* foreground (depicted in red), which makes sense in that the two foregrounds are the girls with similar skin and hair colors but their main difference lies in their clothes.

4.2. ImageNet Dataset

Dataset: ImageNet [6] may not be a perfect dataset for the evaluation of multiple foreground segmentation because each image contains only a single object class with a significant size. Instead, the main objectives of the evaluation with ImageNet [6] are to show (i) the scalability of our method, and (ii) the performance evaluation for the single foreground cosegmentation as a simplified task.

Baselines: We follow the experiment setting of [11] in order to compare our segmentation performance with those of (COS) [11], (LDA) [18], and MNcut [4] that are reported in [11]. We select 50 synsets that provide bounding box labels, and apply our technique to 1000 randomly selected images per synset in both supervised (MFC-S) and unsupervised (MFC-U) ways. In (MFC-S), the foreground models are initialized from the labels of 50 randomly chosen images. Finally, we compute segmentation accuracies by using the provided bounding box annotations.

Results: Fig.5 shows the segmentation accuracies for 13 selected synsets. The accuracies of (MFC-U) and (MFC-S) are higher than those of the best baselines (COS) by more than 3% and 8%, respectively. As discussed before, our algorithm is linear to M and it took about 20 min for 1,000 images on a single machine. We show some selected cosegmentation examples in the supplementary material.

5. Conclusion

We propose the MFC algorithm, a less restrictive and more practical method for multiple foreground cosegmentation. Among future work that could further boost performance, first, one can use a more sophisticated foreground model such as the deformable part model [7] to assess more accurately how valuable a region is to each foreground; second, it is worth exploring other tractable cases of region assignment (*e.g.* relaxing the tree assumption in Section 3.1).

Acknowledgment This research is supported by Google, NSF IIS-0713379, and NSF DBI-0640543.

References

[1] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. Interactively Co-segmentating Topically Related Images with Intelligent Scribble Guidance. *IJCV*, 93:273–292, 2011. 1, 2, 3

[2] Y. Boykov and M.-P. Jolly. Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images. In *ICCV*, 2001. 3

[3] C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE trans. Information Theory*, 14:462–467, 1968. 5

[4] T. Cour, F. Benezit, and J. Shi. Spectral Segmentation with Multiscale Graph Decomposition. In *CVPR*, 2005. 7

[5] P. Cramton, Y. Shoham, and R. Steinberg. *Combinatorial Auctions*. The MIT Press, 2005. 3, 4

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 2, 6, 7

[7] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part-Based Models. *IJCV*, 32:1627–1645, 2010. 3, 7

[8] D. S. Hochbaum and V. Singh. An Efficient Algorithm for Co-segmentation. In *ICCV*, 2009. 1, 2, 3

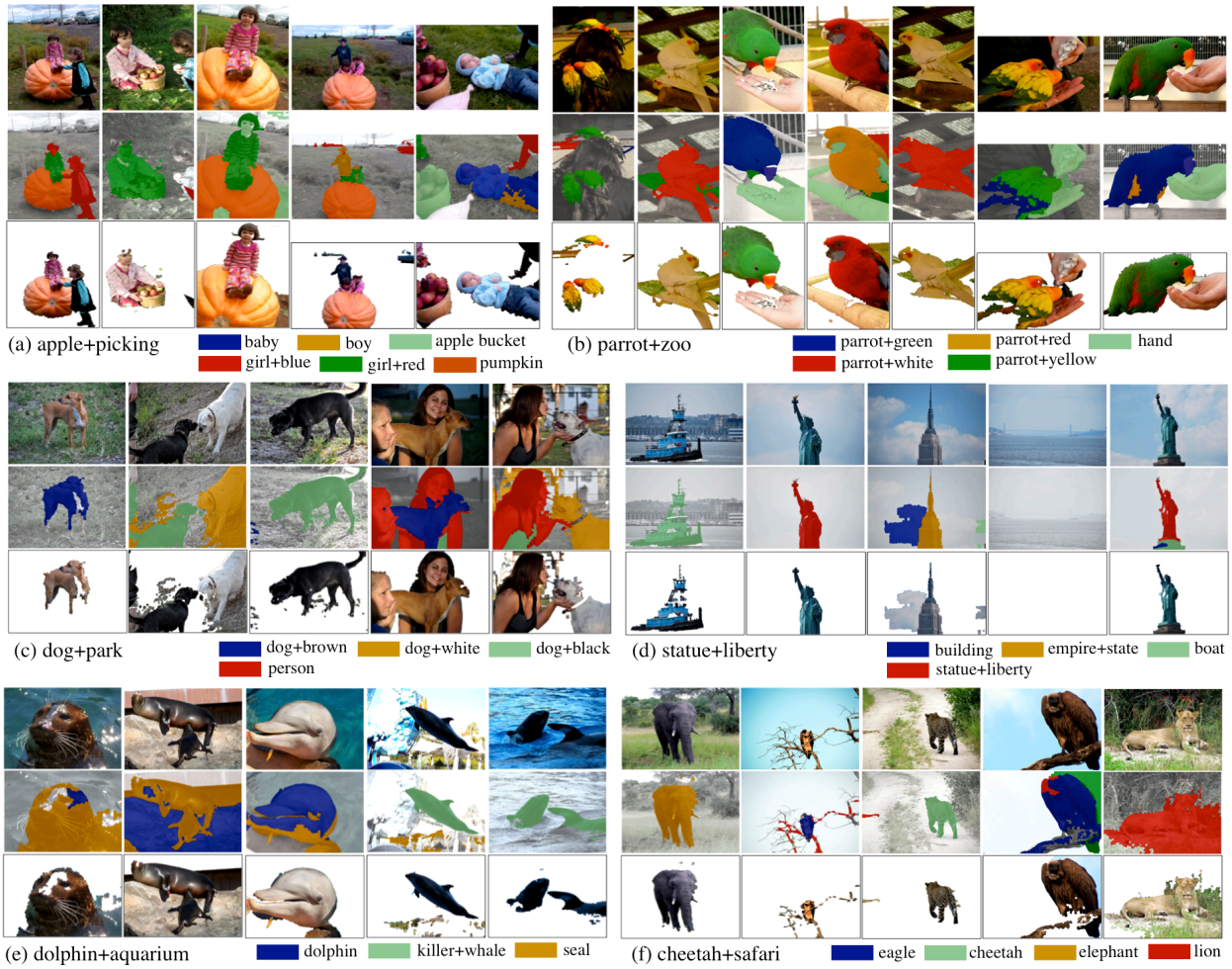

Figure 6. Examples of multiple foreground cosegmentation on some selected groups of FlickrMFC dataset. We sampled 5∼7 images per group. Each set presents input images, color-coded cosegmentation output, and segmented foregrounds, from top to bottom. The color bars below each set indicate which foregrounds are assigned to colored regions.

[9] A. Joulin, F. Bach, and J. Ponce. Discriminative Clustering for Image co-segmentation. In *CVPR*, 2010. 1, 2, 3, 6

[10] G. Kim and A. Torralba. Unsupervised Detection of Regions of Interest Using Iterative Link Analysis. In *NIPS*, 2009. 2

[11] G. Kim, E. P. Xing, L. Fei-Fei, and T. Kanade. Distributed Cosegmentation via Submodular Optimization on Anisotropic Diffusion. In *ICCV*, 2011. 1, 2, 3, 4, 6, 7

[12] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Efficient Subwindow Search: A Branch and Bound Framework for Object Localization. *PAMI*, 31:2129–2142, 2009. 3

[13] S. Lazebnik, C. Schmid, and J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *CVPR*, 2006. 3

[14] L. Mukherjee, V. Singh, and J. Peng. Scale Invariant Cosegmentation for Image Groups. In *CVPR*, 2011. 2, 3

[15] C. Rother, V. Kolmogorov, and A. Blake. GrabCut – Interactive Foreground Extraction using Iterated Graph Cuts. In *SIGGRAPH*, 2004. 2, 3, 6

[16] C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of Image Pairs by Histogram Matching Incorporating a Global Constraint into MRFs. In *CVPR*, 2006. 1, 2, 3

[17] O. Russakovsky and A. Y. Ng. A Steiner Tree Approach to Efficient Object Detection. In *CVPR*, 2010. 3

[18] B. Russell, A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006. 2, 6, 7

[19] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2009. 5

[20] T. Sandholm and S. Suri. BOB: Improved Winner Determination in Combinatorial Auctions and Generalizations. *Artificial Intelligence*, 145:33–58, 2003. 5

[21] T. Sandholm, S. Suri, A. Gilpin, and D. Levine. CABOB: A Fast Optimal Algorithm for Winner Determination in Combinatorial Auctions. *Manage. Sci.*, 51:374–390, 2005. 5

[22] S. Vicente, V. Kolmogorov, and C. Rother. Cosegmentation Revisited: Modes and Optimization. In *ECCV*, 2010. 1, 2, 3

[23] S. Vicente, C. Rother, and V. Kolmogorov. Object Cosegmentation. In *CVPR*, 2011. 1

[24] S. Vijayanarasimhan and K. Grauman. Efficient Region Search for Object Detection. In *CVPR*, 2011. 3