Analyzing the Subspace Structure of Related Images: Concurrent Segmentation of Image Sets*

Lopamudra Mukherjee[†] Vikas Singh[§] Jia Xu[§] Maxwell D. Collins[§]

[†]University of Wisconsin-Whitewater [§]University of Wisconsin-Madison
mukherjl@uww.edu, vsingh@biostat.wisc.edu, {jiaxu,mcollins}@cs.wisc.edu

Abstract. We develop new algorithms to analyze and exploit the joint subspace structure of a set of related images to facilitate the process of concurrent segmentation of a large set of images. Most existing approaches for this problem are either limited to extracting a single similar object across the given image set or do not scale well to a large number of images containing multiple objects varying at different scales. One of the goals of this paper is to show that various desirable properties of such an algorithm (ability to handle *multiple* images with *multiple* objects showing *arbitary* scale variations) can be cast elegantly using simple constructs from linear algebra: this significantly extends the operating range of such methods. While intuitive, this formulation leads to a hard optimization problem where one must perform the image segmentation task together with appropriate constraints which enforce desired algebraic regularity (e.g., common subspace structure). We propose efficient iterative algorithms (with small computational requirements) whose key steps reduce to objective functions solvable by *max-flow* and/or nearly *closed form identities*. We study the qualitative, theoretical, and empirical properties of the method, and present results on benchmark datasets.

1 Introduction

Image segmentation is among the most widely studied problems in the computer vision community. The classical setting, which is how this problem is generally formalized in the literature, is *unsupervised*: one assumes that the underlying model requires no user involvement. While a completely automated solution still remains the de-facto objective, given the difficulty (and ill-posedness) of the task, in recent years we have seen a small but noticeable shift towards interactive image segmentation methods [1]. The goal here is to segment *a given image* with only nominal user interaction. Clearly, obtaining the best segmentation for *one* image is important – but we must note that the proliferation of massive image sharing platforms have created a significant shift in how image data typically presents itself. Images today are rarely generated as independent samples, but rather manifest as 'collections'. Since shared content is pervasive in such sets, modern algorithms must clearly go beyond the analysis of one image at a time.

* This work is supported via NIH R21AG034315, NSF RI 1116584, NSF CGV 1219016, UW-ICTR and W-ADRC. M.D.C. was supported by the UW CIBM Program.

Fig. 1. A set of images with two actors showing quasi-independent scale variations.

This strategy already works well in image categorization and object recognition problems [2], where leveraging large training corpora of images for the learning task is common. On the image segmentation front, the multiple image focused developments are relatively more recent and fall under the umbrella term of Cosegmentation [3]. The premise of Cosegmentation is that when many images containing the same foreground object are available, such shared content may be able to much reduce the need for user guidance [4].

Cosegmentation refers to segmenting a "similar" object from a set of images jointly, with an additional global constraint which forces the foreground appearance models to be similar. Both the unsupervised and the supervised versions of the problem have been actively studied in the last few years [4,5,6,7,8,9,10,11]. On the unsupervised side [3,8], cosegmentation approaches generally operate under the assumption that the background regions in the images are *disparate*: this is essential to rule out the case where the entire image is segmented as the foreground (the appearance models match trivially and the global constraint is less meaningful). Supervised (or weakly supervised) cosegmentation methods [4,10,11,12], on the other hand, address this issue via some interactive user scribble. In conjunction with the choice of appropriate pixel-wise features and/or wrapper inference methods, these models account quite well for changes in illumination, shape and scale variations, and reliably segment an object of interest from multiple images jointly. However, note that this body of work primarily addresses the setting where the set of images contains *a single object of interest*. Heuristic modifications aside, the core mathematical justification behind most existing models [11,7,4,13] does *not* carry through to multiple objects unless we make the impractical assumption that the scale of all objects varies identically across the image set. We show an illustrative example and discuss these details shortly.

Consider the set of images in Fig. 1 which we wish to segment jointly. These images consist of two actors (a dog and a deer), where each exhibits substantial scale changes depending on how close it is to the camera. In some images, one of the actors is temporarily occluded or not in the field of view (i.e., scale is zero). This example is not atypical – a surprisingly large number of image sets (including many instances in the popular iCoseg dataset [4]) consist of more than a single object of interest which co-occur across the image set. Viewing this as a multi-class Cosegmentation entails running the model for each class, one by one, which is often cumbersome if user interaction is needed. This is also an impediment in adapting Cosegmentation in analyzing video data. The algorithms described in this paper are motivated by some of these issues.

The main **contribution** of this paper is to make Cosegmentation approaches applicable to a significantly more general setting. Rather than ask that the fore-

grounds 'share' a parametric (or non-parametric) model [4], impose rank deficiency of the matrix of object appearances [13], or compare images pairwise [3,7] **(a)** we propose new formulations to identify the subspace(s) spanned by a small set of basis appearance models that can best reconstruct the entire set of composite foregrounds (pertaining to multiple objects) in the images. For such a strategy to work, three key components, namely, i) sparse basis subset selection, ii) subspace reconstruction, and iii) image segmentation must happen in tandem. This leads to an interesting (albeit difficult) optimization model. **(b)** We show how effective solutions can be derived for both the supervised and unsupervised versions based on subspace clustering, sparse representation methods and the theory of *maximizing* submodular functions. This provides an elegant framework which permits general non-parametric appearance model *compositions*, that is, the foreground may include tens of objects, at arbitrary scales.

2 Related Work

Initial methods for cosegmentation performed figure-ground labeling of a given pair of images, and enforced a matching (mutual consistency) requirement on the appearance models of the foreground. Various objectives and solution strategies have since been proposed (see [8] for a technical summary), and shown to work well when the number of images is limited to two. This special case is restrictive, and more recent works have extended the ideas to multiple image segmentation. The first step was taken by [4] which suggested constructing a shared mixture model to encode the appearance of a similar foreground object in all images. As noted by [8], this algorithm also shares the background model across the given set of images – a potential problem when the images do not have a substantial shared baseline. Vicente [8] proposed a solution to this problem for the two image setting. Contemporary to these results, [9] identified a nice relationship of Cosegmentation with maximum margin clustering, but the method is computationally quite expensive (especially for a large number of images). Chu [5] showed a small set of results using a method which looks for common patterns in a pre-processing step. Recently, [13] and [11] presented multi-image formulations of the problem. While [13] performs a sequence of iterations involving a segmentation step followed by a rank decomposition of the appearance model matrix, [11] scores similarities between a large set of proposal segmentations. But neither framework is directly generalizable to the *multi-object instances* in the iCoseg dataset or the type of examples shown in Fig. 1. Finally, a few recent papers have incorporated co-saliency [14], used cosegmentation for image classification [15], and extended the algorithms for the cosegmentation of shapes (see [16] for an example of this line of work). Table 1 summarizes the state of the art for the problem to place the contribution of this paper in context.

3 Subspaces of Multiple Object Foreground

Most existing cosegmentation literature performs joint segmentation of all images and simultaneously regularizes the objective based on coherence among

Article	≥ 2 objects	Images	Objective function	Solution Method
Rother [3]	No	2	Graph-cuts plus ℓ_1 norm	Trust-region method
Mu [6]	No	2	Quadratic energy plus generative model	Markov Chain Monte Carlo
Mukherjee [7]	No	2	Graph-cuts plus ℓ_2 norm	Linear Program
Vicente [8]	No	2	Graph-cuts plus generative model	EM like procedure
Hochbaum [10]	No$^{\text{seebelow}}$	2	Joint segmentation with similarity reward	Pseudoflow
Batra [4]	No	Multiple	Graph-cuts plus GMM	Iterative Graph-cuts
Vicente [11]	No	Multiple	Similarity of proposal segmentation pairs	Graph-cuts, A* inference, Random forests
Joulin [9]	No$^{\text{seebelow}}$	Multiple	Discriminative clustering	Convex relaxation of SDP
Mukherjee [13]	No	Multiple	Graph-cuts with a rank one constraint	Iterative network flow and SVD
Chang [14]	No	Multiple	Graph cuts with saliency prior	Graph cuts
This work	**Yes**	**Multiple**	—	—

Table 1. State of the art for Cosegmentation; note that [5] is not included above because that method runs an offline common pattern discovery, and then adjusts a unary term in the segmentation. [9] is potentially applicable to multi-class segmentation, but is computationally expensive, cf. [9], section 3.2. and so only one object case was tackled. Hochbaum [10] does not seem straightforward to adapt for multiple objects. We very recently learnt of works [17,18] which detects multiple objects. These works are not discussed and evaluated here.

the segmented foreground appearance models of the respective images. Assume that $E_{\text{seg}}(\cdot)$ denotes an appropriate segmentation energy (summed over all images), and $C(\cdot)$ is the cosegmentation regularizer which expresses a measure of coherence among the foreground appearance models of the images provided. For example, [3] and [4] use a MRF energy for $E_{\text{seg}}(\cdot)$ and a mixture model based penalty for $C(\cdot)$, but various other options have also been proposed. Since the common building block of our algorithms is the subspace structure of similar foreground regions across images, it seems natural to approach this problem by identifying special forms of $C(\cdot)$ that offer this behavior.

Our first task is to decide on an appropriate representation (i.e., description) for the objects or foregrounds within the images. For both the object-level appearance model as well as the descriptor of the entire foreground, we make use of a visual dictionary over textons (very similar to the object recognition literature [19]). Filter bank responses, when clustered, provide a "texton histogram" where cluster centers with their corresponding covariances define a *visual word* (or a histogram bin). Distinct objects correspond to distinct distributions over k texton bins [12]. Based on this construct, assume that the histograms of each unique object which may appear in the images are provided as $\{m_1, \cdots, m_d\}$ for d objects, where for an object l, $m_l \in \mathbb{R}^k$. With this definition, it follows directly that the foreground in each to be segmented image (say, $f^{[i]}$ in image i) must be a vector in $\mathbb{R}^k$, and can be expressed as $f^{[i]} = \alpha_1 m_1 + \ldots + \alpha_d m_d$ (note that we are operating on the same set of dictionary of visual words or texton bins). Clearly, $\alpha_l = 0$ implies that the l-th object is missing in the i-th image and $\alpha_l > 0$ gives a scaled version of the object-wise texton histogram. This discussion does not yield an implementable algorithm yet (because neither the object-wise texton histogram nor the foreground regions are known).

The Subspace Structure of Foregrounds. Denote the set of foreground appearance vectors for s images as $\{F(:,1),\cdots,F(:,s)\} = \{f^{[1]},\cdots,f^{[s]}\}$. Let us consider a simple example (two objects, three images) to see the subspace structure by focusing on the three respective foregrounds, $f^{[1]}$, $f^{[2]}$, and $f^{[3]}$, assuming that the object models in these foregrounds are indexed by m_1 and m_2. We have $f^{[1]} = \theta_1 m_1 + \theta_2 m_2$, $f^{[2]} = \theta_3 m_1 + \theta_4 m_2$, and $f^{[3]} = \theta_5 m_1 + \theta_6 m_2$ for some set of constants $\{\theta_1,\cdots,\theta_6\}$. Observe that the three foregrounds share the same basis in m_1 and m_2, and so we may write $f^{[3]}$ as a linear combination of $f^{[1]}$ and $f^{[2]}$. Also, $f^{[1]}$ is expressable by combining $f^{[2]}$ and $f^{[3]}$, and similarly $f^{[2]}$ in terms of $f^{[1]}$ and $f^{[3]}$ (a change of basis argument). Denote the coefficients of these linear combinations by a matrix, C whose (j,i)-th entry denotes the contribution of foreground $f^{[j]}$ in expressing $f^{[i]}$. So, the requirement that *every* foreground appearance model should be expressable as a linear combination of a set of basis texton histograms can be achieved by asking that each $f^{[i]}$ (individual columns of F) must be reconstructable as *a linear combination of all other* $f^{[j]}$ where $j \neq i$ ($f^{[i]}$ does not contribute in its own reconstruction). This can be written as $F = FC$ with the condition that the diagonal entries of C must be identically zero, i.e., $\mathrm{diag}(C) = 0$ (where $F \in \mathbb{R}^{k\times s}$ and $C \in \mathbb{R}^{s\times s}$). If the columns of F lie in the *same subspace*, this constraint is satisfied. However, the linear form also permits the identification of *multiple* subspaces into which the columns of F can be 'clustered'. The latter interpretation is strongly related to recent developments in subspace clustering [20,21]. Finally, to permit small variations in the appearance models and make the model robust, we have $F = \hat{F} + \zeta$ where F is composed of a main component $\hat{F}$ plus a noise matrix ζ.

As a final ingredient, we also need to algebraically express the foreground vectors $F(:,i)$ as a function of the segmentation. For each image, we have the texton histogram of the entire image where rows (and columns) correspond to histogram bins (and image pixels) respectively. We denote this as a binary matrix $Z^{[i]}$, where $Z^{[i]}(b,p) = 1$ implies pixel p is assigned to visual word b (like the similarity indicator used in [10]). Let the unknown segmentation indicator variable for image i be $\mathbf{x}^{[i]}$. Then, each entry of $Z^{[i]}\mathbf{x}^{[i]}$ is the dot product of a row a in $Z^{[i]}$ with $\mathbf{x}^{[i]}$, and provides the number of pixels from bin a assigned to foreground. So, $Z^{[i]}\mathbf{x}^{[i]} = F(:,i) = f^{[i]}$. With these components, multi-object multi-image scale free cosegmentation takes the simple form as in (1):

$$\begin{aligned} \min_{\mathbf{x},C,\zeta} \quad & \sum_i E_{\mathrm{seg}}(\mathbf{x}^{[i]}) + \|\zeta\|^2 \\ \text{subject to} \quad & \mathrm{diag}(C) = 0, \quad \mathrm{rank}(C) \leq \kappa \text{ (a small constant).} \\ & F = \hat{F} + \zeta, \quad \hat{F} = \hat{F}C, \quad Z^{[i]}\mathbf{x}^{[i]} = F(:,i), \end{aligned} \tag{1}$$

where the rank constraint offers a regularization on C, with similar motivation as in the subspace clustering literature [20]. The non-convex rank constraint is replaced by its convex relaxation: the nuclear norm. We will ensure fidelity between F and $\hat{F} + \zeta$ as well as between $\hat{F}$ and $\hat{F}C$ as soft constraints by penalizing their respective differences in the objective. The constraint $\hat{F} = \hat{F}C$ is a seemingly difficult quadratic form of two matrix *variables*. But even when

included in the objective, it has a suprisingly simple solution because of the structure of C, as described shortly.

For concreteness of the presentation below, we now decide on the form of $E_{\text{seg}}(\mathbf{x}^{[i]})$ in (1). In this paper, we use the Markov Random Field segmentation, popular for a variety of computer vision applications, see [22]. Other linear forms are possible as long as the the optimal *real*-valued solution can be found in polynomial time in Step 2 below. The main descent steps of the optimization are:

1) Choose a matrix $\hat{F}$ based on some initialization (e.g., the matrix of all ones).

2) With $\hat{F}$ given, optimize $\min_{\mathbf{x}} \quad \sum_i E_{\text{seg}}(\mathbf{x}^{[i]}) + \|F - \hat{F}\|^2 \quad \text{s.t} \quad \mathbf{x} \in [0,1], \quad$ to recover $\mathbf{x}$. We do not solve for C since $\hat{F}$ is given. Using $\mathbf{x}$, calculate each column of F as $Z^{[i]}\mathbf{x}^{[i]}$.

3) Then, optimize (2) to recover $\hat{F}$ and C,

$$\min_{\hat{F},C} \quad \gamma_1\|F - \hat{F}\|^2 + \gamma_2\|\hat{F} - \hat{F}C\|^2 + \|C\|_* \quad \text{s.t.} \qquad \text{diag}(C) = 0 \qquad (2)$$

keeping F fixed. $\|C\|_*$ is nuclear norm. The user specified constants γ_1, γ_2 penalize the soft constraints.

4) Repeat Steps 2–3 until convergence (or negligible change in solution).

Properties. It turns out that the core of the procedure (Step 2 and Step 3) can be performed very efficiently. Let us first analyze Step 2. When E_{seg} is MRF, Step 2 with $\mathbf{x} \in [0,1]$ is a Quadratic Pseudoboolean function (for which fast implementations are already available). Interestingly, Step 3 also turns out to be very easily solvable as shown by [21] (cf. Lemma 2). In fact, in Step 3, the solution of $\hat{F}$ and C such that it satisfies the constraints above can be obtained from a singular value decomposition of F. Since both steps are optimally solvable, we obtain the following simple result:

Lemma 1. *The objective value of the relaxed version of (1) is non-increasing with each iteration.*

Beyond Lemma 1, convergence to a stationary point requires making use of the *persistence* property from [23,24] to show that the set of solutions is finite. Then, the stationary point statement follows by arguments similar to results for convergence of k-means, as shown in [13].

4 Supervised Cosegmentation with Dictionaries of Appearance Models

The preceding model, while interesting, needs discriminative backgrounds across the given image set. This criteria is not satisfied in many datasets depicting multiple objects, where some images may be temporally related and therefore share a common background. This issue does not have an easy solution in the unsupervised setup, but can be addressed effectively by endowing the model with some form of weak supervision to make the problem well posed.

Consider a situation where the user interacts with the model on a few images in the set (the level of supervision is comparable to a GrabCut type scribble interaction [4]), which is then used to derive an approximate texton-based appearance model of the objects of interest. We call this setup cosegmentation with a *precise* dictionary. Note that 'precise' refers not to the quality of the appearance model, rather the fact that the dictionary consists *only* of appearance models of objects likely to appear in the set. We also study a more general version of the problem: it assumes the availability of a larger *overcomplete* dictionary made up of a diverse (and redundant) collection of appearance models. We give a brief overview of the precise dictionary version next, and then discuss its extensions.

Given a small collection of approximate appearances of objects as vectors (distributions over texture visual words), $\mathbf{M} = \{m_1, \cdots, m_d\}$, we want to segment the foreground from unseen images (where objects may appear at arbitrary scales). This problem can be written out as follows (γ is a constant):

$$\min_{\mathbf{x}^{[i]},\lambda} \quad E_{\text{seg}}(\mathbf{x}^{[i]}) + \gamma \|F(:,i) - \sum_{m_j \in \mathbf{M}} \lambda_j m_j\|^2 \quad \text{s.t.} \qquad F(:,i) = Z^{[i]}\mathbf{x}^{[i]}, \quad \mathbf{x}^{[i]} \in [0,1]. \tag{3}$$

The objective penalizes the difference of the unknown foreground $F(:,i)$ (for a fixed i) from a linear combination of the given basis vectors (object appearances). Since $\mathbf{M}$ is known, this problem can be solved very efficiently for the MRF objective as well as other segmentation functions considered in [25]. For instance, if we use MRFs for segmentation, we can obtain provably partially optimal solutions. To do this, we first substitute the basis set $\mathbf{M}$ with an orthogonal basis $\mathbf{M}'$ (using Gram-Schmidt). Then, the penalty term $\gamma\|F - \sum_{m_j \in \mathbf{M}} \lambda_j m_j\|^2$ is interpretable as the *distance* of the vector $F(:,i)$ to the subspace spanned by the vectors in $\mathbf{M}$ or the orthogonal set $\mathbf{M}'$. The advantage of using $\mathbf{M}'$ is that such a distance can be computed in closed form by projecting $F(:,i)$ on to this subspace expressing it as a linear combination of its projection to the orthogonal basis vectors. That is, $\text{proj}_{\mathbf{M}'}(F(:,i)) = \sum_{m_j \in \mathbf{M}'} \lambda_j m_j$ where $\lambda_j = \frac{F(:,i)\cdot m_j}{m_j \cdot m_j}$. For any image i in a given set, the objective function, therefore, takes the form,

$$\min_{\mathbf{x}^{[i]}} \quad E_{\text{seg}}(\mathbf{x}^{[i]}) + \gamma_1 \|F(:,i) - \text{proj}_{\mathbf{M}'}(F(:,i))\|^2, \tag{4}$$

which can be written as a Pseudoboolean function [23] in $\mathbf{x}$, and permits network flow-based solutions. Next, we build upon the ideas above, where the final optimization core will solve a problem similar in form to (4) as a module.

4.1 Cosegmentation with Overcomplete dictionary

Knowledge of precisely which basis vectors will be used in representing the unknown foreground regions, while useful as a first step, restricts applicability of Cosegmentation in several circumstances. For example, consider a temporal image sequence consisting of two main actors (objects), as shown in Fig. 1. In some of the frames, one of the actors may be outside the field of view; therefore, it is not a good idea to use *all* available basis vectors to segment *every* image in the given set. Rather, we would like the algorithm to identify the *smallest* subset

of bases that can be linearly combined to define the foreground of the images (restricting the model complexity). Further, such dictionaries are not difficult to construct using datasets such as MSRC Object Categories, Pascal VOC, and iCoseg using just weak supervision. Once a large universe of approximate object appearance models is available, the goal is to cosegment a given set of images, where the foreground is composed of *a small subset* of appearance models A from our dictionary, D. This problem shares similarities to the *dictionary selection* problem in [26,27,28], but with salient differences. In [26], the goal is to identify a sparsifying sub-dictionary by selecting dictionary columns from multiple candidate bases, and then representing the signal as a sparse reconstruction of the chosen bases.

$$\min_{\mathbf{x}^{[i]},\lambda} \quad \sum_i E_{\text{seg}}(\mathbf{x}^{[i]}) + \gamma_1 \sum_i \|F(:,i) - \sum_{m_j \in A, A \subseteq D, |A| \leq \beta} \lambda_j m_j\|^2 \tag{5}$$

$$\text{s.t.} \quad \forall i \quad F(:,i) = Z^{[i]}\mathbf{x}^{[i]}, \quad \mathbf{x}^{[i]} \in [0,1]. \tag{6}$$

But here, the to-be-reconstructed vector F is *not* fixed, rather needs to be solved in conjuction with other terms. Further, finding the sparse representation standalone is insufficient; instead, it needs to interact with $E_{\text{seg}}(\mathbf{x}^{[i]})$ [1].

Combinatorial Properties. If we use MRF for $E_{\text{seg}}(\cdot)$, in the current setup it is a submodular function [29]. So, we focus on the second part of the objective and define the following function: $L(F(:,i), A) = \|F(:,i) - \sum_{m_j \in A} \lambda_j m_j\|^2$. Note that, given F, the subset of D which best approximates it, can be written as $\hat{A} = \arg\min_{A \in D, |A| \leq \beta} \sum_i L(F(:,i), A)$. Let ϕ be the null set. We define an additional function $G(F(:,i), D) = L(F(:,i), \phi) - \min_{A \in D, |A| \leq \beta} L(F(:,i), A)$ which reduces variance between the linear combination of the chosen bases and the signal to be approximated. This function, when maximized also provides an equivalent sparse representation of the signal. It turns out that such a function is *approximately* sub-modular (see [26]) and its 'deviation' from submodularity is a function of the maximum *incoherency* $\mu = \max_{\forall u,v,u \neq v} \langle m_u, m_v \rangle$. With these tools in hand, we can directly make the following observation.

Observation 1 *The model in (5) can be expressed in the form:* $\min E - G$*, where* E *(same as* E_{seg}*) is submodular,* G *is approximately submodular (*$-G$ *is approximately super-modular), and so* $E - G$ *is a sum of submodular and (approximately) supermodular terms.*

Next, we show how the sub-supermodular function approximation method proposed by [30] can be extended to our problem. To do this, we substitute the supermodular term with its (approximately) modular approximation. This function is defined, wrt to a fixed subset A, as $\Psi(F(:,i), A) = L(F(:,i), \phi) - L(F(:,i), A)$,

[1] The choice of extracting $A \subset D$ instead of regularizing the ℓ_1-norm of λ was driven by empirical feedback. Using a Lasso penalty (relaxation of ℓ_0 norm) involves solving a linear program which may become a bottleneck in vision applications. Second, while penalizing large values in λ (a consequence of ℓ_1) has the undesirable effect of making the model less immune to scale changes, giving unsatisfactory performance.

and can be shown to be approximately modular (see [26]). In our model, the important advantage is that the term $E - \Psi$ can replace the objective $E - G$, which is now approximately submodular (a sum of submodular and approximately modular terms). In addition, it is similar in form to (3), since when the set A is fixed, the problem reduces to a precise dictionary setup. Therefore, efficient methods from §4 are directly applicable. Based on these properties, we adopt the following iterative procedure:

1) Solve the function E and get an initial estimate for $F_{[t]}$ (t refers to the iteration number).
2) Solve $A_{[t]} = \arg\max_{A \subseteq D} G(F_{[t]}, D)$. This can be done using the procedure described in [26]. Note that since $G(F_{[t]}, D) = \psi(F_{[t]}, A_{[t]})$, we have $E - G(F_{[t]}, D) = E - \psi(F_{[t]}, A_{[t]})$.
3) Solve the optimization problem $\min_{\mathbf{x}} E - \psi(:, A_{[t]})$ keeping $A_{[t]}$ fixed, using a procedure similar to §4. Denote the optimal solution by $\mathbf{x}_{[t+1]}$ and the matrix of new foreground vectors as $F_{[t+1]}$.
4) Repeat Steps 2–3 until convergence (or negligible change in solution).

We can now prove the following result:

Proposition 1. *The objective function value is monotonically non-increasing with the iterations.*

Proof (sketch). Note that after Step 3, we get

$$F_{[t]} - G(F_{[t]}, D) = E_{[t]} - \psi(F_{[t]}, A_{[t]}) \geq E_{[t+1]} - \psi(F_{[t+1]}, A_{[t]}).$$

This is because as we are solving the optimization problem in Step 3 to optimality. Further,

$$E_{[t+1]} - \psi(F_{[t+1]}, A_{[t]}) \geq E_{[t+1]} - G(F_{[t+1]}, D).$$

This is true because in Step 2, $A_{[t+1]} = \arg\max_{A \subseteq D} G(F_{[t+1]}, D)$; therefore, $G(F_{[t+1]}, D) \geq \psi(F_{[t+1]}, A_{[t]})$; otherwise replacing $A_{[t+1]}$ by $A_{[t]}$ improves the solution of $G(F_{[t+1]}, D)$ trivially and the solution converges. Therefore, we directly have $E_{[t]} - G(F_{[t]}, D) \geq E_{[t+1]} - \psi(F_{[t+1]}, A_{[t]}) \geq E_{[t+1]} - G(F_{[t+1]}, D)$, and so the iterations either decrease the objective value at each step or the iterations converge.

Generating class specific labels: The reader will notice that while our algorithms identifies multiple objects at arbitary scale variations in a set of images, the output is in the form of a *joint* foreground indicator vector, rather than class specific indicator vectors. But class specific indicators can be obtained from such an output, if desired. The main task is to divide the joint foreground indicator vector $F(:, i)$ of image i, into the constituent class specific indicator vectors. To do this, we project the foreground indicator vector $F(:, i)$ on to the basis vectors, to obtain foreground appearance model for each object individually (say $F_d(:, i)$ for object class d). We can then decompose the indicator vector $\mathbf{x}^{[i]}$, into an indicator vector for each object class $\mathbf{x}_d^{[i]}$, satisfying the property that they agree with the object-wise models above, i.e., $Z^{[i]}\mathbf{x}_d^{[i]} \simeq F_d(:, i)$. This is essentially a

least squares problem of the form $Ax \simeq b$. It turns out the the LHS coefficient matrix (A) of this form has a totally unimodular property, therefore if we round the RHS (b) to integral values, such a least squares problem will have an exact solution.

5 Evaluations

Our experiments were designed to assess the model's performance on several benchmark datasets, using existing methods as a baseline. Broadly, the setup consists of: evaluation of (a) the unsupervised algorithms in §3, and (b) the supervised algorithms with exact and overcomplete dictionaries in §4 – §4.1. We demonstrate some examples for the unsupervised model, but mainly focus our attention to the more broadly applicable methods from Section 4.1, which were evaluated on the entire iCoseg dataset [4] and a subset of MSRC object categories . In addition, we also include comparison of our supervised method with fully supervised SVM. We used texture-based appearance models as described in Section 3 using agglomerative information bottleneck from [19]. The unary terms for the MRF objective were created using the GMMs from the Grabcut implementation in OpenCV using the training data (when available) or by specifying a box centered on the image covering 60% area (in the unsupervised setting). All segmentations were done at the pixel level (no superpixels were used).

Subspace Cosegmentation of Multiple Objects. We performed a preliminary evaluation of this model using a small number of examples collected from the internet.

Since the algorithm assumes that only the foreground regions are similar (and the background is disparate), we extracted images from several video sequences which were temporally separated. Representative examples (from Toy Story) are shown in Fig. 2 where there is significant pose/shape variation in the objects; further two of the images consist of only one character. The model performs favorably relative to [9] (also an unsupervised approach).

Fig. 2. Results of the algorithm in Section 3 (Row 2) relative to segmentation obtained from [9] (Row 3).

Cosegmentation with appearance dictionaries. These experiments are a rigorous assessment of the model because the dataset includes deformable objects, and significant variations in pose, viewpoint, as well as scale. Interestingly, not all images contain all objects which allow properly evaluating all properties of our algorithm in §4.1.

ICoseg. The iCoseg dataset contains 38 image categories with up to 40 images in each class. For each class, we created a small training set consisting of

up to 2 training examples (from the ground truth) to generate the dictionary (this can also be derived from scribble guidance [1]). We illustrate comparisons of our approach with three other methods from [11], [8], and [9]. Among these the cosegmentation method of [11] uses training data but by a very different procedure. Since the performance of any cosegmentation method varies among different classes, similar to other papers [4] we report the results for each class. Also, consistent with common practice [11,4,13], we report accuracy as the percentage of pixels in the image (both foreground and background) which were correctly classified. (note that results in [11] included a subset of all images in each class). Since the model decomposes into independent runs in §4.1, it is not limited by how many images can be segmented at once. In Table 2, we summa-

Fig. 3. Some results of the model in §4.1 on multi-object Liverpool (cols 1-4) and Soccer sets (cols 5-8)

rize our accuracy summaries after segmenting all ~ 640 images from all classes in iCoseg. Overall, compared to the accuracy numbers reported for each class in [11] (and also [8], [9]), our model performs well and yields better accuracy in all but two classes. Some visual results are presented in Figure 3 to illustrate its qualitative performance on images with multiple objects (including scenes where an object is missing). Note that for the Liverpool and the Women Soccer images shown, the ground truth provided in iCoseg only asks for detecting one object. To detect all objects, we created a dictionary with only one training example for each team (by running a Grabcut with a few scribbles, and retaining results from the first iteration). Even though the training examples were not perfect, the results in Fig. 3 indicate the algorithm can identify multiple objects with relative ease, and is mostly immune to situations where one or more objects are not visible in a scene.

class	Ours	[11]	[8]	[9]	class	Ours	[11]	[8]	[9]
Balloon	**95.17%**	90.10%	89.30%	85.20%	Kite Panda	**93.37%**	90.20%	70.70%	73.20%
Baseball	**95.66%**	90.90%	69.90%	73.0%	Panda	**92.83%**	92.70%	80.00%	84.00%
Brown bear	88.52%	**95.30%**	87.3%	74.0%	Skating	**96.64%**	77.50%	69.9%	82.1%
Elephants	**87.65%**	43.10%	62.3%	70.1%	Statue	**96.64%**	93.80%	89.3%	90.6%
Ferrari	**89.95%**	89.90%	77.7%	85.0%	Stonehenge1	**92.67%**	63.30%	61.1%	56.6%
Gymnastics	**92.18%**	91.70%	83.4%	90.9%	Stonehenge2	84.87%	**88.80%**	66.9%	86.0%
Kite	**94.63%**	90.3%	87.0%	87.0%	Taj Mahal	**94.07%**	91.1%	79.6%	73.7%

Table 2. Segmentation accuracy summaries for image classes from iCoseg dataset.

MSRC Object Categories. The MSRC dataset contains several categories of objects, but in each object class, the constituent images are much more diverse

Approach	Sheep	Car	Cow	Flowers	Plane	Dog	Bird
Ours	89.0%	**80.1%**	87.8%	86.5%	**87.1%**	**93.5%**	94.8%
[11]	**93.0%**	79.6%	**94.2%**	–	83.0%	93.1%	**95.3%**

Fig. 4. Segmentation accuracy on MSRC.

compared to ICoseg. For example, the Flowers class includes flowers of different colors and shapes: in such cases, for cosegmentation to yield very high accuracy, far richer visual features may be needed. To make our models applicable, we created a dictionary having one representative image of each unique type which provided $4-5$ training examples per class – all other images in the class were then presented to the model for segmentation. The accuracy is summarized for the subset of classes tested are shown in Fig. 4 using the recent work of [11] (which also used training) as a baseline. Overall, this suggests that the performance of our algorithm is similar to [11]. Finally, we observe that both methods are limited only by the underlying visual features that enable (a) comparing proposal segmentations in [11] and (b) comparing appearance descriptors in ours. Examples from MSRC and iCoseg is shown in Fig. 5.

Results on comparison with fully supervised SVM. Since the algorithms described in Sections 4 and 5 are essentially supervised, we compare our method with a fully supervised algorithm such as SVM. SVMs were run on images from the ICoseg dataset, since the background and foregrounds are both fixed for such images. For each image group, we select five images as the training set (note that for experiments using our method, we used no more than two training image). For each training image, we compute a texton feature descriptor (17 features) for each pixel and train a classifier based on that (we use the built-in svmtrain function in Matlab with SMO as the solver). After that, we use the learned classifier and test it on the remaining image set. Figure 6 shows some representative images. In general, the results of SVM are about $10-15\%$ worse than our method and also worse than any other baseline used in the main paper. This is somewhat expected as our algorithm imposes an appearance model constraint on the entire set of pixels labeled as foreground by asking that they span a subspace given by a subset of known appearances. But similar patches routinely co-occur in the foreground and background, which throws off the results of SVM substantially in the absence of any terms that make the solution behave like a valid segmentation (e.g., homogeneity).

Other Comments. Our results above show that the model yields results that are superior or competitive with the state of the art on two benchmark datasets. The run-time increases near linearly with each image; the main cost is minimizing

Fig. 5. Results of the algorithm in §4.1 on the ICoseg (cols 1-5) and MSRC (cols 6-8)

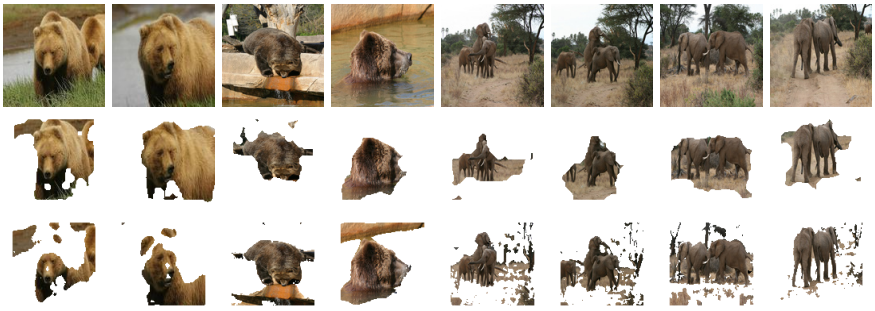

Fig. 6. Results of the comparison of our algorithm with fully supervised SVM on three datasets from Icoseg: Rows 1 shows the original images, Rows 2 shows the results of our approach and Rows 3 shows the results using SVM

a QPB function which takes $5 - 20s$ per image per iteration (convergence in 5 iterations). Other than these experiments, we evaluated how often the "correct" basis vectors $A \subset D$ are chosen by the algorithm during segmentation. To do this, we manually found correspondences between each image in the test and training class for MSRC data. The number of histogram bin centers in [19] was fixed to 500. Feedback for MSRC experiments suggested that for the 125 images in Fig. 4, the model identified the correct basis subset over 90% of the time.

6 Discussion

We propose new algorithms for simultaneous segmentation of multiple objects from image collections, by analyzing and exploiting their shared subspace structure. Our models, for both unsupervised and supervised setting, extend the current state of the art for such approaches, which until now, has been limited to identifying a single common object. We believe this makes idea of cosegmentation applicable to a much wider class of problems, therefore significantly extends the operating range of such methods. Experiments on benchmark datasets show that algorithm performs well on a variety of image sets.

References

1. Rother, C., Kolmogorov, V., Blake, A.: "GrabCut": interactive foreground extraction using iterated graph cuts. ACM Trans. on Graphics **23** (2004) 309–314
2. Gehler, P.V., Nowozin, S.: On feature combination for multiclass object classification. In: ICCV. (2009)
3. Rother, C., Minka, T., Blake, A., Kolmogorov, V.: Cosegmentation of image pairs by histogram matching: Incorporating a global constraint into MRFs. In: CVPR. (2006)
4. Batra, D., Kowdle, A., Parikh, D., Luo, J., Chen, T.: iCoseg: Interactive co-segmentation with intelligent scribble guidance. In: CVPR. (2010)
5. Chu, W., Chen, C., Chen, C.: MOMI-cosegmentation: Simultaneous segmentation of multiple objects among multiple images. In: ACCV. (2010)

6. Mu, Y., Zhou, B.: Co-segmentation of image pairs with quadratic global constraint in MRFs. In: ACCV. (2007)
7. Mukherjee, L., Singh, V., Dyer, C.R.: Half-integrality based algorithms for cosegmentation of images. In: CVPR. (2009)
8. Vicente, S., Rother, C., Kolmogorov, V.: Cosegmentation revisited: Models and optimization. In: ECCV. (2010)
9. Joulin, A., Bach, F., Ponce, J.: Discriminative clustering for image co-segmentation. In: CVPR. (2010)
10. Hochbaum, D.S., Singh, V.: An efficient algorithm for co-segmentation. In: ICCV. (2009)
11. Vicente, S., Kolmogorov, V., Rother, C.: Object cosegmentation. In: CVPR. (2011)
12. Collins, M.D., Xu, J., Grady, L., Singh, V.: Random walks based multi-image segmentation: Quasiconvexity results and gpu-based solutions. In: CVPR. (2012)
13. Mukherjee, L., Singh, V., Peng, J.: Scale invariant cosegmentation for image groups. In: CVPR. (2011)
14. Chang, K., Liu, T., Lai, S.: From cosaliency to cosegmentation: An efficient and fully unsupervised energy minimization model. In: CVPR. (2011)
15. Chai, Y., Lempitsky, V., Zisserman, A.: Bicos: A bi-level co-segmentation method for image classification. In: ICCV. (2011)
16. Glasner, D., Vitaladevuni, S., Basri, R.: Contour based joint clustering of multiple segmentations. In: CVPR. (2011)
17. Kim, G., Xing, E.P.: On multiple foreground cosegmentation. In: CVPR. (2012)
18. Joulin, A., Bach, F., Ponce, J.: Multi-class cosegmentation. In: CVPR. (2012)
19. Winn, J., Criminisi, A., Minka, T.: Object categorization by learned universal visual dictionary. In: ICCV. (2005)
20. Liu, G., Lin, Z., Yu, Y.: Robust subspace segmentation by low-rank representation. In: ICML. (2010)
21. Favaro, P., Vidal, R., Ravichandran, A.: A closed form solution to robust subspace estimation and clustering. In: CVPR. (2011)
22. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. PAMI **23** (2001)
23. Boros, E., Hammer, P.: Pseudo-Boolean optimization. Disc. Appl. Math. **123** (2002) 155–225
24. Rother, C., Kolmogorov, V., Lempitsky, V., Szummer, M.: Optimizing binary mrfs via extended roof duality. In: CVPR. (2007)
25. Hochbaum, D.: Polynomial time algorithms for ratio regions and a variant of normalized cut. PAMI **32** (2010)
26. Cevher, V., Krause, A.: Greedy dictionary selection for sparse representation. IEEE J. of Selected Topics in Signal Processing **5** (2011) 979–983
27. Jenatton, R., Mairal, J., Obozinski, G., Bach, F.: Proximal methods for sparse hierarchical dictionary learning. In: ICML. (2010)
28. Krause, A., Cevher, V.: Submodular dictionary selection for sparse representation. In: ICML. (2010)
29. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? PAMI **26** (2004)
30. Narasimhan, M., Bilmes, J.: A submodular-supermodular procedure with applications to discriminative structure learning. In: UAI. (2005)