Semi-supervised Learning for Large Scale Image Cosegmentation

Zhengxiang Wang Rujie Liu
Fujitsu Research & Development Center Co., Ltd, Beijing, China
{wangzhengxiang,rjliu}@cn.fujitsu.com

Abstract

This paper introduces to use semi-supervised learning for large scale image cosegmentation. Different from traditional unsupervised cosegmentation that does not use any segmentation groundtruth, semi-supervised cosegmentation exploits the similarity from both the very limited training image foregrounds, as well as the common object shared between the large number of unsegmented images. This would be a much practical way to effectively cosegment a large number of related images simultaneously, where previous unsupervised cosegmentation work poorly due to the large variances in appearance between different images and the lack of segmentation groundtruth for guidance in cosegmentation.

For semi-supervised cosegmentation in large scale, we propose an effective method by minimizing an energy function, which consists of the inter-image distance, the intra-image distance and the balance term. We also propose an iterative updating algorithm to efficiently solve this energy function, which decomposes the original energy minimization problem into sub-problems, and updates each image alternatively to reduce the number of variables in each subproblem for computation efficiency. Experiment results on iCoseg and Pascal VOC datasets show that the proposed cosegmentation method can effectively cosegment hundreds of images in less than one minute. And our semi-supervised cosegmentation is able to outperform both unsupervised cosegmentation as well as fully supervised single image segmentation, especially when the training data is limited.

1. Introduction

The problem of image cosegmentation is actively studied in recent computer vision community. Given a set of related images with the prior knowledge that they all contain a common object, the goal of cosegmentation is to automatically find this common object in each image and segment it as foreground. This problem is firstly proposed in [24], which serves as a special case of figure-ground segmentation compared to single image segmentation. The original cosegmentation studies [24, 19, 20, 11, 26] could only handle just a pair of images. Recent studies [13, 5, 27, 21, 14, 25, 22] extend this limitation and can cosegment multiple images. This is an important progress to cosegmentation, and it makes this research more practical for real world problem since there are usually more than two images in reality that share a common object. However, the size of image set in these studies is still limited to only dozens of images. When it contains hundreds of images, current methods may either work poorly due to the dramatically increased variances in appearance between different images, or run slowly due to the expensive computation cost. [17, 15] have tried to cosegment hundreds of or even thousands of images, but they use clustering strategy that divides the large image set into multiple subsets, and then cosegment each subset separately. This may not be an optimal solution as it avoids to directly cosegment the whole image set, and the similarity information (about the common object) between images in different subsets is lost.

In this paper, we try to cosegment a large number of images simultaneously, which is a much challenging task due to the large variance between different images. If some training image foregrounds are provided, it is possible to guide the cosegmentation task towards a correct direction. However, due to the expensive cost of human labeling, the training data is usually very limited, and result in limited accuracy by traditional supervised single image segmentation. Therefore, we introduce to use semi-supervised cosegmentation, which can outperform both unsupervised cosegmentation (here “unsupervise” refers to without using any training segmentation groundtruth, although all images are known to contain the common objects in assumption) and supervised single image segmentation in this case, because it exploit the similarity from both the segmented foregrounds in training images, as well as the common object shared between different unsegmented images.

We propose an effective method for this semi-supervised cosegmentation, which minimizes the energy function consisting of the inter-image distance, the intra-image distance and the balance term. The inter-image distance measures the similarity of foregrounds between pairwise images. It uses training image foregrounds for guidance in cosegmentation, and exploits the similarity from both the training images and unsegmented images. The intra-image distance considers spatial continuity within each unsegmented image. And the balance term prevents segmenting the whole image as foreground or background. With these three terms, the resulting energy minimization problem can be formulated as a binary quadratic programming (QP) problem, which is able to effectively segment the foreground of each unsegmented image.

Efficiency is also a very important issue in cosegmenting a large number of images. To increase efficiency, we propose an iterative updating algorithm using the trust region idea to solve the energy function. That is, we update every image one by one alternatively in each iteration, by keeping the foregrounds of other images fixed and updating the foreground of each image as a sub-problem. This updating iteration is repeated until convergence. Compared to updating all images simultaneously using only one iteration, this iterative updating algorithm can significantly reduce the number of variables in each sub-problem and therefore speed up the whole procedure. In each sub-problem, we also approximate the binary QP problem by a continuous convex QP problem for fast computation. For cosegmenting hundreds of images, only less than one minute is required by using this iterative updating algorithm.

After solving the above mentioned accuracy and efficiency issues in cosegmenting a large number of images, our proposed semi-supervised method is more practical for real world applications than previous cosegmentation works. We summarize our contributions in this paper as follows:

- We firstly introduce a semi-supervised cosegmentation task, which makes use of both the limited training segmentation groundtruth, as well as the common object shared between different unsegmented images, for large scale image cosegmentation.
- We propose an effective method for semi-supervised cosegmentation by minimizing an energy function that consists of the inter-image distance, the intra-image distance and the balance term.
- We propose an efficient algorithm to solve the energy function by iterative updating, which is able to cosegment hundreds of image in less than one minute.

We organize the rest of this paper as follows. Section 2 briefly reviews previous studies related to our work. We describe our energy function and the iterative updating algorithm in Section 3, and evaluate its performance in Section 4. Finally, we conclude this paper in Section 5.

2. Related Work

Image Cosegmentation: The problem of cosegmentation is firstly proposed by Rother et al. [24], in which a common object shared by two images is segmented by measuring the similarity between their foreground histograms with L1-norm. The resulting foregrounds by cosegmentation would be helpful in many other applications. For example, Rother et al. [24] show that the distance between an image pair measured by the cosegmented foregrounds can help improve image retrieval. Chai et al. [3, 4] use the cosegmented image foregrounds to successfully help improve the performance of image classification.

Due to its usefulness in other computer vision applications, cosegmentation has been actively studied in recent years. [20, 11, 26] try to use other measuring approaches to compare the two foreground histograms for cosegmentation. Recent works [13, 5, 27, 21, 25] extend previous limitation of cosegmenting only two images, and can cosegment multiple images. The work in [17, 14, 22] also extend the foreground-background binary segmentation to multiple regions, which is able to cosegment multiple images with multiple objects. Another recent work in [16] tries to cosegment with multiple foregrounds, which would be a more challenging problem. All these works are unsupervised and limited to cosegment at most dozens of images simultaneously. For segmenting large scale dataset, [17, 15] use clustering strategy to divide the large image set into multiple subsets, and cosegment each subset separately. [18] transfers segmentations from segmented images in the current source set to unsegmented images in the next target set by segmentation propagation, and finally segment the whole ImageNet dataset [8]. However, these works do not cosegment all images simultaneously, and may lose the similarity information between images in different subsets.

Semi-Supervised Learning: Semi-supervised learning is especially useful when the training data is limited and there are plenty of unlabeled data [6]. It is actively studies in machine learning and surveyed in [28]. In computer vision, semi-supervised learning is mainly used in image classification [10] and retrieval [12]. For cosegmentation, most previous works use unsupervised learning as mentioned before, while there are also some supervised learning methods. The transductive segmentation method proposed in [7] try to transduce the cutout model to other related images for object cutout. Batra et al. [1] uses user scribble guidance to segment images and then recommend to users where to scribble next. These methods are unable to effectively and efficiently cosegment a large number of images, which would be benefit by semi-supervised learning. To the best of our knowledge, we are the first to introduce semi-supervised learning for large scale image cosegmentation.

3. Methodology

Given N_s training images with segmentation groundtruth and N_u unsegmented images, suppose all these images contain the common object as the prior knowledge, and this common object is labeled as foreground in each training image, the task of semi-supervised cosegmentation is to find this common object in every unsegmented image and label it as foreground.

For this task, superpixels are firstly extracted from each image in pre-processing, so that the foreground/background label can be defined on each superpixel rather than on each pixel for computation efficiency. For each training image, the label of each superpixel can be easily determined by comparing the areas covered by foreground and background. For each unsegmented image, this task is formulated as predicting the label for each superpixel, then the final foreground can be found by selecting superpixels with foreground labels.

A vector y_i is used to represent the superpixel labels for an image X_i, with the dimension s_i equal to the number of superpixels in this image. Each component $y_i(j)$ in vector y_i is a binary variable, with 1 indicating the corresponding superpixel j belongs to foreground and 0 for background. The determination of y_i for each unsegmented image is formulated as an energy function minimization problem, which is then solved by an iterative updating algorithm.

3.1. Cosegmentation energy function

Before giving the definition of the energy function, we first give some notations. Like many previous works [24, 20, 11, 26, 21, 5, 15] , histogram descriptors are used to represent superpixels and the foregrounds of images, which can be either bag-of-words histogram with some local features, or color histogram based on pixel intensities. The superpixel histogram is represented by $h_i(j) \in R^d$ for each superpixel j in image X_i, and the foreground histogram of image X_i can be calculated as $\sum_j y_i(j) \cdot h_i(j)$, which can also be formulated as $H_i \cdot y_i$, where H_i is a $(d \times s_i)$ matrix with each column corresponding to $h_i(j)$.

3.1.1 Energy function definition

The proposed energy function is composed of three terms: the inter-image distance, the intra-image distance and the balance term, in which all unsegmented images are included. Therefore by solving the minimizing problem with this energy function, the superpixel labels of all unsegmented images can be calculated simultaneously.

The **inter-image distance** measures the similarity of foregrounds between different images, including the similarity between unsegmented images and training images as well as that between pair-wise unsegmented images. Therefore both the training segmentation groundtruth and the similarity information shared between unsegmented images are explored in the inter-image distance. The Euclidean distance is used to compare two foreground histograms as in [20], then the corresponding energy function is formulated as:

$$\begin{aligned} E_{inter} &= \sum_{i=1}^{N_u}\sum_{j=1}^{N_s} \parallel H_i \cdot y_i - H_j^{tr} \cdot y_j^{tr} \parallel^2 \\ &+ \sum_{i=1}^{N_u}\sum_{j=i+1}^{N_u} \parallel H_i \cdot y_i - H_j \cdot y_j \parallel^2 \end{aligned} \tag{1}$$

where H_j^{tr} and y_j^{tr} refers to superpixel histograms and labels for training images respectively.

The **intra-image distance** considers the spatial consistency between two adjacent superpixels inside an unsegmented image. This term tries to assign the same label to visually similar adjacent superpixels, i.e., foreground or background, by adding a penalty to the energy function in case that two adjacent superpixels are given different labels. Therefore the corresponding energy function is formulated as:

$$E_{intra} = \sum_{i=1}^{N_u} \sum_{j=1,k=1}^{s_i} W_i(j,k) \cdot \delta(j,k) \tag{2}$$

where $\delta(j,k)$ measures whether two superpixels j and k have different labels and is defined as:

$$\delta(j,k) = \{ \begin{array}{ll} 1, & if \;\; y_i(j) \neq y_i(k) \\ 0, & if \;\; y_i(j) = y_i(k) \end{array} = |y_i(j) - y_i(k)| \tag{3}$$

$W_i(j,k)$ is the penalty term measuring the edge affinity of two superpixels j and k. It is defined in a similar form as in [15] if j and k are adjacent:

$$W_i(j,k) = \frac{\alpha(j,k)}{\sum_{l \in N(j)} \alpha(j,l)} \cdot \exp(-\frac{\parallel h_i(j) - h_i(k) \parallel^2}{\theta}) \tag{4}$$

or 0 in case they are not adjacent. Here $\alpha(j,k)$ is the shared edge length between two superpixels j and k, $N(j)$ is the set of adjacent superpixels of j, and θ is a constant value, which is set as the variance of the distance values between all superpixel histograms.

The **balance term** prevents all superpixels belonging to the same label during the energy minimization procedure. The entropy of the proportion of foreground and background superpixels is used to measure this term:

$$E_{bal} = \sum_{i=1}^{N_u} (P_i^f \log P_i^f + P_i^b \log P_i^b) \tag{5}$$

where the proportion of foreground superpixels P_i^f is measured as:

$$P_i^f = \frac{\sum_{j=1}^{N_u} y_i(j)}{s_i} = \frac{y_i^T \cdot e_i}{s_i} \tag{6}$$

where e_i is a vector with the same dimension to y_i and all components equal to 1. The proportion of background superpixels P_i^b can be calculated by $(1 - P_i^f)$.

By summing these three terms, **the whole energy function** can be formulated as:

$$E = E_{inter} + \lambda_1 \cdot E_{intra} + \lambda_2 \cdot E_{bal} \tag{7}$$

where λ_1 and λ_2 are two trade-off parameters to control the proportion of each term in the energy function.

3.1.2 Binary quadratic programming problem

Given the definition of the energy function, the minimization can be converted to a binary QP problem, by reformulating each of the three terms into suitable form. Due to the limitation of space, detailed derivation is put in the supplementary material and here we directly present the reformulated result.

The **inter-image distance** in Equation 1 can be reformulated to:

$$\begin{aligned} E_{inter} \quad = \quad & \sum_{i=1}^{N_u} y_i^T \cdot M_{ii}^{inter} \cdot y_i & (8) \\ + \quad & \sum_{i=1}^{N_u} \sum_{j=i+1}^{N_u} y_i^T \cdot M_{ij}^{inter} \cdot y_j + \sum_{i=1}^{N_u} y_i^T \cdot V_i + C \end{aligned}$$

where M_{ii}^{inter} is a $(s_i \times s_i)$ matrix calculated as:

$$M_{ii}^{inter} = (N_u + N_s - 1) \cdot H_i^T \cdot H_i \tag{9}$$

M_{ij}^{inter} is also a $(s_i \times s_i)$ matrix calculated by:

$$M_{ij}^{inter} = -2H_i^T \cdot H_j \tag{10}$$

V_i is a vector with dimension of s_i:

$$V_i = -2H_i^T \cdot \sum_{j=1}^{N_s} H_j^{tr} \cdot y_j^{tr} \tag{11}$$

Since the superpixel label y_j^{tr} of training images are known, it can be treated as a constant vector during the minimization procedure.

C is a scalar calculated by:

$$C = N_u \cdot \sum_{i=1}^{N_s} (y_i^{tr})^T \cdot (H_i^{tr})^T \cdot H_i^{tr} \cdot y_i^{tr} \tag{12}$$

It is also a constant value and has no effect on the minimization result, therefore it can be omitted during the minimization procedure.

The **intra-image distance** in Equation 2 can be reformulated to:

$$E_{intra} = \sum_{i=1}^{N_u} y_i^T \cdot M_i^{intra} \cdot y_i \tag{13}$$

where M_i^{intra} is a $(s_i \times s_i)$ Laplacian matrix. Its diagonal component $M_i^{intra}(j,j)$ is calculated as:

$$M_i^{intra}(j,j) = \sum_{k \in N(j)} (W_i(j,k) + W_i(k,j)) \tag{14}$$

and the off-diagonal component $M_i^{intra}(j,k)$ is calculated as follows if superpixel j and k are adjacent, or 0 otherwise.

$$M_i^{intra}(j,k) = -W_i(j,k) - W_i(k,j) \tag{15}$$

The **balance term** in Equation 5 can be approximated to the following form through Taylor expansion:

$$E_{bal} = \sum_{i=1}^{N_u} (2 \frac{y_i^T \cdot e_i \cdot e_i^T \cdot y_i}{s_i^2} - 2 \frac{y_i^T \cdot e_i}{s_i} - \frac{1}{2}) \tag{16}$$

Therefore **the whole energy function** E can be reformulated to the following form after omitting all constant scalars:

$$\begin{aligned} E = & \sum_{i=1}^{N_u} y_i^T \cdot (M_{ii}^{inter} + \lambda_1 M_i^{intra} + \lambda_2 \frac{e_i \cdot e_i^T}{s_i^2}) \cdot y_i & (17) \\ + & \sum_{i=1}^{N_u} \sum_{j=i+1}^{N_u} y_i^T \cdot M_{ij}^{inter} \cdot y_j + \sum_{i=1}^{N_u} y_i^T \cdot (V_i - \lambda_2 \frac{e_i}{s_i}) \end{aligned}$$

By concatenating all superpixel labels of unsegmented images into a long vector Y, the above function can be formulated to the following binary QP problem:

$$\min_Y E = Y^T \cdot M \cdot Y + Y^T \cdot V \tag{18}$$

where M is a large matrix, its diagonal block M_{ii} corresponding to image i is:

$$M_{ii} = M_{ii}^{inter} + \lambda_1 M_i^{intra} + \lambda_2 \frac{e_i \cdot e_i^T}{s_i^2} \tag{19}$$

and the off-diagonal block M_{ij} corresponding to image i and j is equal to $\frac{1}{2} M_{ij}^{inter}$. V is a long vector concatenating vectors of the value $(V_i - \lambda_2 \frac{e_i}{s_i})$ corresponding to each image i.

3.2. Iterative updating algorithm

The binary QP problem has been studied extensively in the optimization literature [2, 23, 20], and Equation 18 can be easily solved using these methods when cosegmenting a small number of images. However, for large scale cosegmentation, as the number of superpixels in all images (the dimension of Y in Equation 18) is increased to a large value, the optimization procedure of these methods will be computation expensive. To increase efficiency, we propose an iterative updating algorithm using the trust region idea to solve this problem. The basic idea of this algorithm is to update every unsegmented image one by one alternatively in each iteration, by keeping the superpixel labels of other images fixed in updating the current image, and repeat this iteration until convergence. In this way, updating the superpixel labels of each image is decomposed as a sub-problem, where the number of variables (superpixel labels) is significantly reduced and the optimization procedure can be accelerated. In updating the superpixel labels y_i of image X_i, the sub-problem E_i is converted from Equation 17 to the following formula (see supplementary material for detail):

$$\min_{y_i} E_i = y_i^T \cdot M_i' \cdot y_i + y_i^T \cdot V_i' + C_i' \tag{20}$$

where

$$M_i' = M_{ii}^{inter} + \lambda_1 M_i^{intra} + \lambda_2 \frac{e_i \cdot e_i^T}{s_i^2} \tag{21}$$

$$V_i' = \sum_{j=1, j\neq i}^{N_u} M_{ij}^{inter} \cdot y_j + V_i - \lambda_2 \frac{e_i}{s_i} \tag{22}$$

C_i' represents the rest terms in Equation 17 that are not related to y_i, which can be omitted as a constant scalar, since the superpixel labels y_j of other unsegmented images are fixed during the minimization of this sub-problem. It can be seen from Equation 20 this sub-problem is also a binary QP problem with significantly reduced number of binary variables compared to Equation 18, and can be easily solved using previous binary QP methods [2, 23, 20]. In the experiment of this paper, we simply relax the binary variable of each superpixel label $y_i(j)$ from $\{0, 1\}$ to $[0, 1]$. Then each sub-problem is approximated as a convex QP problem since each M_i' is positive semi-definite (this can be easily verified from its definition, but is omitted in this paper due to the limitation of space), and can be solved in polynomial time using general QP solver such as active set. The resulting value is then rounded to binary value for superpixel labels.

In the iterative updating algorithm, all sub-problems are solved individually to update the superpixel labels of the corresponding images. In two successive iterations, the only difference in updating each image X_i of sub-problem E_i is that the labels of other unsegmented images y_j would be changed, therefore only the first term ($\sum_{j=1, j\neq i}^{N_u} M_{ij}^{inter} \cdot y_j$) in vector V_i' (Equation 22) of each sub-problem is required to be re-calculated. As this term needs to sum over all other images, the complexity of updating all images grows quadratically with the number of images, which seems inefficient for large scale cosegmentation. However, this calculation can be further accelerated from $O(N_u)$ to $O(1)$ and improve the updating algorithm with linear complexity. This is because according to Equation 10, the re-calculated term can be rewritten as:

$$\begin{aligned} \sum_{j=1, j\neq i}^{N_u} M_{ij}^{inter} \cdot y_j &= -2H_i^T \cdot \sum_{j=1, j\neq i}^{N_u} H_j \cdot y_j \\ &= -2H_i^T \cdot (S - H_i \cdot y_i) \end{aligned} \tag{23}$$

where

$$S = \sum_{j=1}^{N_u} H_j \cdot y_j \tag{24}$$

is a summation term kept throughout the whole updating procedure. After getting a new superpixel label vector y_i^{new} in the updating of image X_i, we also need to update S by:

$$S^{new} = S^{old} + H_i \cdot (y_i^{new} - y_i^{old}) \tag{25}$$

Since Equation 23 and 25 require only $O(1)$ complexity, the whole updating procedure can be improved to linear complexity and makes the large scale cosegmentation much efficient.

Another advantage of the iterative updating algorithm is that it can also reduce the rounding error compared to directly solving energy function of Equation 18 (where the superpixel labels of all images need to be rounded simultaneously). This is because the rounding error of superpixel labels only occurs in the corresponding sub-problem and will be fixed in other sub-problems. Therefore the final objective value E by iterative updating algorithm can be more close to the actual optimal minimum.

The proposed iterative updating algorithm is similar to trust region graph cut in [24]. As indicated in [24], this method requires a good initialization for segmentation in the first iteration. For unsupervised segmentation, this is indeed a difficult problem. However, in our semi-supervised cosegmentation, the limited training images provide a good initialization and can guide the cosegmentation towards a correct direction for unsegmented images. Moreover, each sub-problem is approximated as a convex QP problem, which makes the initialization for unsegmented images not important anymore. We simply set all initial superpixel labels as 1.

The trade-off parameters λ_1 and λ_2 have to be tuned empirically in unsupervised cosegmentation, which is also a problem as in previous studies [5]. However, in the semi-supervised setting, these two parameters can be tuned automatically with the training segmentation groundtruth. Nevertheless, our proposed method can also be used for unsupervised cosegmentation, by simply removing V_i in Equation 22 and setting N_s to 0 for each sub-problem.

4. Experiment

We use iCoseg [1] and Pascal VOC 2012 [9] datasets to evaluate the proposed method. iCoseg dataset is popularly used in previous cosegmentation works [1, 27, 25, 14],

Table 1. Cosegmentation accuracy comparison in iCoseg dataset

Classes	Ours	Joulin 2010	Kim 2011	Kim 2011	Joulin 2012	Joulin 2012
		[13]	(Best K) [17]	(K=2) [17]	(Best K) [14]	(K=2) [14]
Baseball	0.592	0.179	0.621	0.123	0.617	0.197
Football	0.463	0.188	0.446	0.176	0.522	0.396
Panda	0.665	0.472	0.517	0.495	0.457	0.340
Goose	0.718	0.745	0.781	0.772	0.795	0.795
Airplane	0.477	0.577	0.054	0.049	0.500	0.146
Cheetah	0.476	0.358	0.614	0.496	0.668	0.636
Kite	0.539	0.651	0.107	0.093	0.532	0.208
Balloon	0.620	0.484	0.465	0.227	0.599	0.298
Statue	0.688	0.907	0.584	0.579	0.887	0.852
Kendo	0.781	0.802	0.716	0.716	0.871	0.709
Average	**0.602**	0.536	0.491	0.373	0.645	0.458

Table 2. Running time comparison (second) in iCoseg dataset

Classes	# images	Ours	Joulin 2010	Kim 2011	Joulin 2012
			[13]	(K=2) [17]	(K=2) [14]
Baseball	25	6.8	963.8	38.6	998.4
Football	33	6.7	1449.4	47.6	1557.4
Panda	25	5.9	1449.6	42.3	941.9
Goose	31	5.9	1028.2	47.6	1050.1
Airplane	39	12.6	1763.8	31.6	1822.5
Cheetah	33	4.6	1533.9	31.5	1642.1
Kite	18	4.1	583.3	20.3	734.3
Balloon	24	2.6	941.2	23.6	829.4
Statue	41	6.0	1257.6	51.6	2018.3
Kendo	30	11.2	2501.8	47.6	1247.9
Average	29.9	**6.6**	1347.3	38.2	1284.2

which contains 38 classes, each for an independent cosegmentation task. However, most classes contain only a few images, therefore we select 10 representative classes containing more images for our cosegmentation experiment, in which the number of images ranges from 18 to 40. The segmentation challenge sets in VOC2012 dataset is originally used for single image segmentation. As it contains the largest number of images with pixel-wise groundtruth labeling inside each class so far as we know, we can also use these images for large scale cosegmentation. For better evaluation and comparing, we select 8 classes with more apparent common objects and consistent scales, with the number of images ranging from 120 to 249 in each class. For the representation of each superpixel and foreground, we use color histogram with RGB and Lab color channels. The intersection-over-union score is used to measure the cosegmentation accuracy, which is a standard evaluation metric in Pascal Challenges [9].

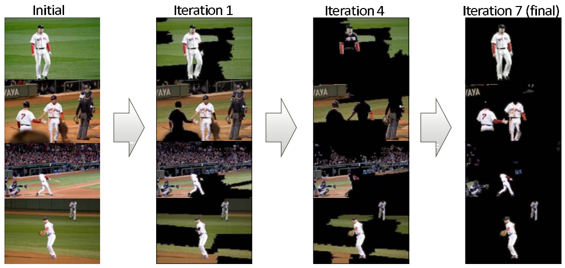

Figure 1. An example showing the intermediate result during the iterative updating algorithm. Note that although only 4 images are shown here as example, this is the intermediate result of cosegmenting all the 25 images in “Baseball” class in iCoseg dataset.

4.1. Cosegmentation results

We first evaluate the unsupervised version of the proposed method. Three recent cosegmentation works [13, 17, 14] are compared in iCoseg dataset, which are implemented by their publicly available code with the default parameter setting. In [17] and [14], images can be cosegmented into multiple regions, therefore we adjust the number of regions K from 2 to 9 and report the best one, for the foreground-background binary cosegmentation in this experiment. The performances of their binary version (when K=2) are also reported. Table 1 shows the cosegmentation accuracy of each class and the average results. By selecting the best K for each class, [14] performs the best in average. However, this comparison is unfair as additional manual work is used to choose the best K. Moreover, it is usually dif-

Table 3. Cosegmentation accuracy comparison in VOC2012 dataset

Classes	Ours	Kim 2011 (Best K) [17]	Kim 2011 (K=2) [17]
Aeroplane	0.335	0.166	0.142
Boat	0.231	0.100	0.098
Bus	0.392	0.342	0.335
Diningtable	0.255	0.228	0.228
Dog	0.248	0.145	0.131
Motorbike	0.280	0.222	0.222
Sheep	0.205	0.148	0.146
Train	0.332	0.220	0.200
Average	**0.285**	0.196	0.188

Table 4. Running time comparison (second) in VOC2012 dataset

Classes	# images	Ours	Kim 2011 (K=2) [17]	Kim 2011 (K=9) [17]
Aeroplane	178	25.8	341.4	1807.3
Boat	150	13.3	348.9	1432.5
Bus	152	15.3	439.9	1631.6
Diningtable	157	11.7	467.6	2225.8
Dog	249	51.7	527.0	2165.1
Motorbike	157	19.4	432.6	1869.6
Sheep	120	34.3	249.0	1142.4
Train	167	15.7	480.3	1898.5
Average	166.3	**23.4**	410.8	1771.6

ficult to determine the best K beforehand in unsupervised cosegmentation tasks. If K is fixed to 2, the result of [14] drops significantly as shown in Table 1. Our method wins in all remained situations, especially [17] with the best K. An example of the intermediate result during our iterative updating algorithm is shown in Figure1, and an analysis of the cosegmentation accuracy affected by the choice of parameters (λ_1 and λ_2) can be found in the supplementary material.

We also compare the running time of these methods. For [17] and [14], only the running time for their binary version is reported since more time is required for multiple regions cosegmentation ($K > 2$). As shown in Table 2, our method only requires 6.6s for cosegmenting 29.9 images in average, which is significantly faster than all the other three methods. It should be noted that the running time shown in this table does not include superpixel extraction and histogram generation steps for all methods.

In VOC2012 dataset, only [17] is compared since it can also cosegment images in large scale. For [13] and [14], the requirement on large memory and computation cost for cosegmenting hundreds of images is beyond our computation capability. The cosegmentation accuracy and running time are presented in Table 3 and 4 respectively. Again our method significantly outperforms [17] for either $K = 2$ or the best K. For cosegmenting hundreds of images, our method only requires less than one minute, which is much more efficient than [17], where about 7 minutes are required for binary segmentation and this value is increased to nearly half an hour when K is set to 9.

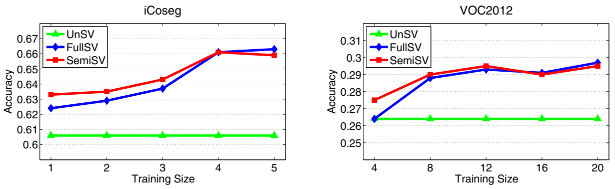

Figure 2. Average result over all classes in iCoseg and VOC2012 datasets.

We also try the cosegmentation experiments at the level of 1000 images. Due to the lack of enough images with groundtruth segmentation in VOC2012 dataset for the accuracy evaluation, we randomly select 1000 related images from its classification challenge set and only test the running time. Our method requires about 5 minutes for cosegmenting 1000 images, while [17] needs $60 - 70$ minutes as reported in their paper. It can be seen that the time complexity of our method is linear with the number of images, which validates our acceleration method.

4.2. Semi-supervised cosegmentation results

Next, the cosegmentation experiment is performed in semi-supervised manner (denoted as "SemiSV") and the result is compared with unsupervised learning (denoted as "UnSV") as well as supervised learning (denoted as "FullSV"). For supervised learning, each image is segmented individually with training images only, without considering the similarity of the common object shared between unsegmented images. It can be easily performed with our energy minimization problem, by removing the term $(M_{ij}^{inter} \cdot y_j)$ in Equation 22 and setting N_u to 0. Besides, one iteration is enough for updating each image, as each image is segmented individually.

Both iCoseg and VOC2012 datasets are used in this experiment, with five groups of different training sizes for each dataset. In iCoseg dataset, 1 to 5 images are randomly selected as the training images in each class for the five groups respectively. The test images for all the five groups are kept the same, chosen from images that are not selected for training in any group. In VOC2012 dataset, as the average number of images is increased to 166.3, the training size is also slightly increased, ranging from 4 to 20. For training image selection in this dataset, we notice that some images have large errors in the superpixel labels, which are determined according to the overlap with the foreground and background pixel labels. That is, the resulting foreground from the converted superpixel labels is significantly different from the original foreground, probably due to bad superpixel extraction. Therefore, instead of random selection, only the images with lower conversion errors are selected for training for better evaluation.

Figure 2 shows the average accuracy of both datasets. It is obvious that "SemiSV" outperforms both "FullSV" and "UnSV" in case of fewer training images. This result shows that semi-supervised learning will be most competent when the number of unsegmented images is far more than that of segmented ones, as concluded in [6]. With the fewest training images in VOC2012 dataset, the accuracy of "FullSV" is close to "UnSV", which indicates that the similarity information from the common object between test images is competitive to that provided by the segmentation groundtruth of the 4 training images in this dataset. With increased training images, the improvement of "FullSV" grows more quickly than "SemiSV". In the group with the most training images, the accuracy of "FullSV" is better than "SemiSV". This is because given the large number of training images, the semi-supervised learning cannot benefit from the common region between test images anymore. What's more, the concrete information from the training images may be stained by the uncertainty of the unsegmented images, which worsens the final cosegmentation accuracy.

5. Conclusion

In this paper, we proposed a semi-supervised learning method for large scale images cosegmentation, where hundreds of images can be processed in less than one minute with competitive cosegmentation accuracy. By making use of both the limited training segmentation groundtruth, as well as the common object shared between the large number of unsegmented images, our semi-supervised cosegmentation method can outperform both unsupervised cosegmentation and supervised single image segmentation, especially when cosegmenting a large number of images with limited training data provided.

References

[1] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *CVPR*, 2010.

[2] A. Billionnet and S. Elloumi. Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem. *Mathematical Programming*, 109(1), 2007.

[3] Y. Chai, V. Lempitsky, and A. Zisserman. Bicos: A bi-level co-segmentation method for image classification. In *ICCV*, 2011.

[4] Y. Chai, E. Rahtu, V. Lempitsky, L. Van Gool, and A. Zisserman. Tricos: A tri-level class-discriminative co-segmentation method for image classification. In *ECCV*, 2012.

[5] K.-Y. Chang, T.-L. Liu, and S.-H. Lai. From co-saliency to co-segmentation: An efficient and fully unsupervised energy minimization model. In *CVPR*, 2011.

[6] O. Chapelle, B. Schölkopf, A. Zien, et al. *Semi-supervised learning*, volume 2. MIT press Cambridge, 2006.

[7] J. Cui, Q. Yang, F. Wen, Q. Wu, C. Zhang, L. Van Gool, and X. Tang. Transductive object cutout. In *CVPR*, 2008.

[8] 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.

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results.

[10] M. Guillaumin, J. Verbeek, and C. Schmid. Multimodal semi-supervised learning for image classification. In *CVPR*, 2010.

[11] D. S. Hochbaum and V. Singh. An efficient algorithm for co-segmentation. In *ICCV*, 2009.

[12] S. Hoi, W. Liu, and S.-F. Chang. Semi-supervised distance metric learning for collaborative image retrieval. In *CVPR*, 2008.

[13] A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, 2010.

[14] A. Joulin, F. Bach, and J. Ponce. Multi-class cosegmentation. In *CVPR*, 2012.

[15] E. Kim, H. Li, and X. Huang. A hierarchical image clustering cosegmentation framework. In *CVPR*, 2012.

[16] G. Kim and E. P. Xing. On multiple foreground cosegmentation. In *CVPR*, 2012.

[17] G. Kim, E. P. Xing, L. Fei-Fei, and T. Kanade. Distributed cosegmentation via submodular optimization on anisotropic diffusion. In *ICCV*, 2011.

[18] D. Kuettel, M. Guillaumin, and V. Ferrari. Segmentation propagation in imagenet. In *ECCV*, 2012.

[19] Y. Mu and B. Zhou. Co-segmentation of image pairs with quadratic global constraint in mrfs. In *ACCV*, 2007.

[20] L. Mukherjee, V. Singh, and C. R. Dyer. Half-integrality based algorithms for cosegmentation of images. In *CVPR*, 2009.

[21] L. Mukherjee, V. Singh, and J. Peng. Scale invariant cosegmentation for image groups. In *CVPR*, 2011.

[22] L. Mukherjee, V. Singh, J. Xu, and M. D. Collins. Analyzing the subspace structure of related images: concurrent segmentation of image sets. In *ECCV*, 2012.

[23] C. Olsson, A. P. Eriksson, and F. Kahl. Solving large scale binary quadratic problems: Spectral methods vs. semidefinite programming. In *CVPR*, 2007.

[24] 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.

[25] J. C. Rubio, J. Serrat, A. López, and N. Paragios. Unsupervised co-segmentation through region matching. In *CVPR*, 2012.

[26] S. Vicente, V. Kolmogorov, and C. Rother. Cosegmentation revisited: Models and optimization. In *ECCV*, 2010.

[27] S. Vicente, C. Rother, and V. Kolmogorov. Object cosegmentation. In *CVPR*, 2011.

[28] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.