Image Co-segmentation via Saliency Co-fusion

Koteswar Rao Jerripothula, Student Member, IEEE, Jianfei Cai, Senior Member, IEEE, and Junsong Yuan, Senior Member, IEEE

Abstract—Most existing high-performance co-segmentation algorithms are usually complex due to the way of co-labelling a set of images as well as the common need of fine-tuning few parameters for effective co-segmentation. In this paper, instead of following the conventional way of co-labelling multiple images, we propose to first exploit inter-image information through co-saliency, and then perform single-image segmentation on each individual image. To make the system robust and to avoid heavy dependence on one single saliency extraction method, we propose to apply multiple existing saliency extraction methods on each image to obtain diverse salient maps. Our major contribution lies in the proposed method that fuses the obtained diverse saliency maps by exploiting the inter-image information, which we call saliency co-fusion. Experiments on five benchmark datasets with eight saliency extraction methods show that our saliency co-fusion based approach achieves competitive performance even without parameter fine-tuning when compared with the state-of-the-art methods.

Index Terms—Saliency, co-saliency, fusion, co-fusion, segmentation, co-segmentation.

I. INTRODUCTION

Image co-segmentation refers to the task of extracting common objects from a set of images, which is very useful for many vision and multimedia applications such as object-based image retrieval, image classification, and object recognition. It can be considered as one type of weakly supervised segmentation methods, which makes use of the weak prior that there exist common objects across different images in the set. This is quite different from single image segmentation. The existing single image object-level segmentation methods can only exploit either the prior from human supervision, which requires human interactions such as GrabCut, or the prior from single image-based visual saliency, which might fail at complex images with cluttered background or non-salient foreground. In contrast, image co-segmentation goes beyond single image segmentation in the sense that it can exploit not only the intra-image priors, but also the inter-image priors. Furthermore, it also brings in the new challenges of how to find the right inter-image priors and how to make use of them.

The concept of co-segmentation was first introduced in [1], which used histogram matching to simultaneously segment out the common object from a pair of images. Since then, many co-segmentation algorithms have been proposed in the literature, ranging from early image pair co-segmentation [2][3], multiple image co-segmentation [4][5][6][7], interactive image co-segmentation [8][9][10] to the recent multiple objects co-segmentation [11][12][13][14], multiple group co-segmentation [15], large-scale co-segmentation [16][17], shape alignment targeted co-segmentation [19] and evaluation criteria driven co-segmentation [20].

Despite the great progress made by the existing co-segmentation algorithms, they still have some major limitations. First, most of the state-of-the-art co-segmentation algorithms require fine-tuning of quite a few parameters and the co-labelling of multiple images simultaneously, which are very complex and time-consuming, especially for large diverse datasets. Second, as seen in the existing works [16][21], co-segmenting images might not perform better than single image segmentation for some datasets. This might be due to the additional energy term commonly used to enforce inter-image consistency, which often results into unsmooth segmentations in individual images.

In this paper, we focus on binary image co-segmentation, i.e. extracting a common foreground from a given image set. Instead of following the conventional way of co-labelling multiple images, we aim to exploit inter-image information through co-saliency, and then perform single-image segmentation on each individual image. Moreover, to make the system robust and avoid heavy dependence on one single saliency extraction method for generating co-saliency, we propose to apply multiple saliency extraction methods on each image.

Eventually, an enhanced saliency map is generated for each image by fusing its various saliency maps via weighted summation at superpixel level, where the weights are optimized by exploiting inter-image information, as shown in Fig. 1. We call the proposed method saliency co-fusion, whose objectives include: (1) boosting the saliency of common foreground re-
contrary, our method is purely an unsupervised approach. A grounds from a given image set [11][12][13][14], where the algorithms being designed for segmenting multiple common foregrounds, have been proposed in the literature. Early approaches [1][2][3] focused on segmenting a pair of images containing one common object with more effective foreground/background priors through similar element voting, and neighbor smoothness constraints. Finally, the fused saliency for a superpixel is just a weighted summation of all the saliency maps of the superpixel. Experimental results show that our saliency co-fusion based co-segmentation achieves effective co-segmentation by sharing shape templates. In [26], co-segmentation problem was addressed by establishing consistent functional maps across images in a reduced functional space, which requires training. Another interesting work [20], which reports state-of-the-art performance, employed region-level matching. Also, it determined a good co-segment by checking whether it can be well composed from other co-segments. Most of these methods focused on pixel level co-labeling whereas we focus on saliency co-fusion.

Our method is closely related to co-segmentation and co-saliency research.

Co-segmentation: Many co-segmentation algorithms have been proposed in the literature. Early approaches [1][2][3] focused on segmenting a pair of images containing one common object. It was later extended to deal with multiple images containing one common object with more effective or more efficient models enforcing inter-image consistency [22][4][5][23][24][25]. However, there are also some algorithms being designed for segmenting multiple common foregrounds from a given image set [11][12][13][14], where the best performers make use of supervised information. On the contrary, our method is purely an unsupervised approach. A few interactive co-segmentation approaches [8][9][10] have also been proposed, where users can give scribbles for one or a small number of the images. Thus, the extracted prior information is then used to influence the segmentation of the entire image set. Our method does not require such human intervention. Recently, [16] applied dense SIFT matching to discover common objects, and co-segment them out from noisy image dataset, where some images do not contain common objects. They tried to enforce inter-image consistency strongly by developing matching based prior, so as to exclude noise image from participating in the co-segmentation process. In [19], co-segmentation was combined with co-sketch for effective co-segmentation by sharing shape templates. In [26], co-segmentation problem was addressed by establishing consistent functional maps across images in a reduced functional space, which requires training. Another interesting work [20], which reports state-of-the-art performance, employed region-level matching. Also, it determined a good co-segment by checking whether it can be well composed from other co-segments. Most of these methods focused on pixel level co-labeling whereas we focus on saliency co-fusion.

Just like we use multiple saliency maps, there are also some methods that use multiple segmentation proposals to perform semantic segmentation. For example, [27] made use of multiple segmentation proposals of an image to come up with several compositions and eventually produce semantic segmentation by searching for high-scoring maximal weighted cliques. Later, [21] extended the idea of using multiple segmentation proposals to the object co-segmentation problem and demonstrated better results than classical co-segmentation algorithms. In this research, in contrast to pre-segmenting and then selecting segmentation proposals, we propose to fuse multiple saliency maps to arrive at an enhanced saliency map and then carry out segmentation.

Co-saliency: Co-saliency typically refers to the common

II. RELATED WORK

Our method is closely related to co-segmentation and co-saliency research. Co-segmentation: Many co-segmentation algorithms have been proposed in the literature. Early approaches [1][2][3] focused on segmenting a pair of images containing one common object. It was later extended to deal with multiple images containing one common object with more effective or more efficient models enforcing inter-image consistency [22][4][5][23][24][25]. However, there are also some algorithms being designed for segmenting multiple common foregrounds from a given image set [11][12][13][14], where the best performers make use of supervised information. On the contrary, our method is purely an unsupervised approach. A few interactive co-segmentation approaches [8][9][10] have also been proposed, where users can give scribbles for one or a small number of the images. Thus, the extracted prior information is then used to influence the segmentation of the entire image set. Our method does not require such human intervention. Recently, [16] applied dense SIFT matching to discover common objects, and co-segment them out from noisy image dataset, where some images do not contain common objects. They tried to enforce inter-image consistency strongly by developing matching based prior, so as to exclude noise image from participating in the co-segmentation process. In [19], co-segmentation was combined with co-sketch for effective co-segmentation by sharing shape templates. In [26], co-segmentation problem was addressed by establishing consistent functional maps across images in a reduced functional space, which requires training. Another interesting work [20], which reports state-of-the-art performance, employed region-level matching. Also, it determined a good co-segment by checking whether it can be well composed from other co-segments. Most of these methods focused on pixel level co-labeling whereas we focus on saliency co-fusion.

Just like we use multiple saliency maps, there are also some methods that use multiple segmentation proposals to perform semantic segmentation. For example, [27] made use of multiple segmentation proposals of an image to come up with several compositions and eventually produce semantic segmentation by searching for high-scoring maximal weighted cliques. Later, [21] extended the idea of using multiple segmentation proposals to the object co-segmentation problem and demonstrated better results than classical co-segmentation algorithms. In this research, in contrast to pre-segmenting and then selecting segmentation proposals, we propose to fuse multiple saliency maps to arrive at an enhanced saliency map and then carry out segmentation.

Co-saliency: Co-saliency typically refers to the common
saliency existing in a set of images containing similar objects. The term “co-saliency” was first coined in [28], in the sense of what is unique in a set of similar images, and the concept was later linked to extract common saliency, which is very useful for many practical applications [29][30]. For example, co-saliency object priors have been efficiently used for co-segmentation in [31]. Recently, a cluster based co-saliency method using various cues was proposed in [32], which learns the global correspondence and obtains cluster saliency quite well. It represents state-of-the-art method due to its simplicity, effectiveness, and efficiency. However, the co-saliency method [32] is mainly designed for images of the same object captured at different viewpoints or instances. It cannot well handle image sets with huge intra-class variation. Another recent work [33] fused saliency maps from different images via warping technique and it is able to handle the intra-class variation. However, most of these methods only use a single saliency map which may not be accurate always.

### III. Saliency Co-fusion

In this section, we first formulate our saliency co-fusion problem. Then we give a detailed description of individual terms as well as implementation details.

#### A. Problem Formulation

Considering a set of \( N \) images \( \mathcal{I} = \{I^1, I^2, \ldots, I^N\} \), denote \( \mathcal{B}^n = \{B^n_1, B^n_2, \ldots, B^n_M\} \) the set of \( M \) saliency maps (normalized to range 0-1) for image \( I^n \) obtained using \( M \) different existing saliency extraction methods. Also, denote \( \mathcal{P}^n = \{P^n_1, P^n_2, \ldots, P^n_{|P^n|}\} \) the set of superpixels in image \( I^n \) obtained using [34]. Defining a saliency map region of superpixel as an element \( e \), which is the basic processing unit in our method, we have total \( N_e = \sum_{n=1}^{N} M|\mathcal{P}^n| \) elements. Let \( z(n, k, m) \) denote the associated weight for element \( e(n, k, m) \) that belongs to image \( n \), superpixel \( k \), and saliency map \( m \). The weight maps depicted in Fig. 1 and Fig. 2 are basically constructed using these associated weights.

We stack all the weights into a vector \( \mathbf{z} = [z_1, z_2, \ldots, z_{N_e}]^t \) for simplicity and use \( u \) or \( v \) as the element indices for referencing purposes. We mix the usage of the element vector index with its corresponding matrix index \((n, k, m)\) since one can be converted to the other easily. Table I summarizes the major notations used throughout the paper.

Our goal is to find the optimal weight for each of the elements in order to jointly fuse various saliency maps of similar images at superpixel level such that common foreground saliency gets boosted up and background saliency is suppressed in final fused saliency maps. In particular, we treat saliency co-fusion as a weight selection problem. On one hand, we want to give higher weights to elements with higher confidence. On the other hand, we want to have certain consistency in the weight selection among neighboring elements. Considering the constraint that the resultant fused saliency map values should occur in the range \([0, 1]\), we formulate our task as a quadratic programming problem:

\[
\begin{align*}
\min_{\mathbf{z}} & \quad D^t\mathbf{z} + \lambda \mathbf{z}^tG\mathbf{z} \\
\text{s.t.} & \quad 0 \leq z_u \leq 1, \quad \forall u \in [1, N_e], \\
& \quad \sum_{m=1}^{M} z(n, k, m) = 1, \quad \forall I^n \in \mathcal{I}, P^n_k \in \mathcal{P}^n
\end{align*}
\]

where there are two terms traded off by a balancing parameter \( \lambda \). The first term \((D^t\mathbf{z})\) is a prior term to enforce global commonness and co-saliency, where the prior term coefficient vector \( D \in \mathbb{R}^{N_e \times 1} \). The second term \((\mathbf{z}^tG\mathbf{z})\) is a pairwise smoothness term to encourage neighborhood elements to take similar weights, where the smoothness term coefficient matrix \( G \in \mathbb{R}^{N_e \times N_e} \). The constraints in Eq. (1) are there to ensure that individual weights range between 0 and 1, and the summation of all the weights for one superpixel is equal to one. Once \( \mathbf{z} \) is determined by minimizing Eq. (1), the fused saliency map \( J^n \) for a pixel \( p \in P^n_k \) can be simply computed as

\[
J^n(p) = \sum_{m=1}^{M} z(n, k, m) \times B^n_{m}(p),
\]

where \( B^n_{m} \) is the \( m \)-th saliency map for image \( I^n \).

#### B. Feature Description and Similarity

Unlike other methods [16][32][33] where features for matching are extracted from images independent of saliency maps, we develop a saliency map based feature descriptor because our processing units are elements (defined as a saliency map region of a superpixel), instead of pixels or superpixels. We consider the fact that there is no uniformity among saliency maps obtained by different methods. For instance, some saliency maps are of high contrast, while others are of poor contrast. Some are bright, while others are dark. This can cause serious problems in the process if saliency values are directly taken as features. We tackle it by distinguishing potential foreground pixels from potential background pixels in an element using the classical Otsu’s method as shown in Fig. 3. For each group (both the potential foreground group and the potential background group in the element), we construct a feature descriptor which consists of the average dense SIFT descriptor, and also the average color values in RGB, HSV, and Lab spaces. However, for each element, we have two feature descriptors with each having dimensions \( d = 128 + 3 + 3 + 3 = 137 \). We concatenate them as the feature descriptor for one element. In this way, different elements of the same superpixel obtain different feature descriptors, depending upon the foreground/background distributions in each element.

\( X_f, X_b \in \mathbb{R}^{N_e \times d} \) and \( X \in \mathbb{R}^{N_e \times 2d} \) denote the data matrices that stack the foreground descriptors, the background descriptors and the foreground/background concatenated descriptors of all the elements as its rows, respectively. We construct similarity matrices \( S_f, S_b \) and \( S \), all of \( N_e \times N_e \) dimensions that record the potential foreground similarity,
the potential background similarity, and the total similarity respectively between all the element pairs:

\[
S_f(u, v) = \exp\left(-\gamma \sum_{q=1}^{d} \frac{(X_f(u, q) - X_f(v, q))^2}{X_f(u, q) + X_f(v, q)}\right) \tag{3}
\]

\[
S_b(u, v) = \exp\left(-\gamma \sum_{q=1}^{d} \frac{(X_b(u, q) - X_b(v, q))^2}{X_b(u, q) + X_b(v, q)}\right) \tag{4}
\]

\[
S(u, v) = \exp\left(-\gamma \sum_{q=1}^{2d} \frac{(X(u, q) - X(v, q))^2}{X(u, q) + X(v, q)}\right) \tag{5}
\]

where \(\gamma\) is a parameter set to \(\frac{1}{500}\).

Note that the potential foreground similarity is set to zero if all the pixels in the element belong to the background group and vice versa. If elements \(u\) and \(v\) belong to the same image, \(S_f(u, v), S_b(u, v),\) and \(S(u, v)\) are all set to zero since we aim at exploiting similar elements from other images.

Based on the total similarity matrix \(S\), similar elements for each element are identified if the corresponding similarity values are larger than a similarity threshold \(\theta\) (\(\theta\) is set to 0.75). For one element, its similar elements provide recommendations via different cues, based on which we then derive the appropriate weight for the considered element so as to encourage or discourage its role in the final fused saliency map. Details are elaborated below.

C. Prior Term

We define our prior term coefficient vector \(D\) in Eq. (1) as

\[
D = D_s + D_f + D_c \tag{6}
\]

which includes three cues: saliency cue \((D_s)\) from similar elements, foreground/background prior cue \((D_f)\) from similar elements and centerness cue \((D_c)\) based on the spatial location of the element.

Saliency Cue: Following the idea of co-saliency or common saliency, we compare the average saliency of similar elements with the average saliency value of the considered element to decide whether the element should be emphasized or not (give high weight or not). Let \(T = [T_1, T_2, ..., T_{N_e}]^T\) denote the vector where each entry is the average saliency value of an
element. On the other hand for an element \( u \), we compute the average saliency recommended by its similar elements as

\[
F_u = \frac{\sum_{v=1}^{N_n} T_v \delta(S(u, v) > \theta)}{\sum_{v=1}^{N_n} \delta(S(u, v) > \theta)}
\]  

(7)

where \( \delta(\cdot) \) is the indication function, equal to one if the condition (\( \cdot \)) is true (otherwise 0), which is used to determine whether element \( v \) is a similar one or not. Let \( F = [F_1, F_2, \ldots, F_{N_n}]^T \) be the vector comprising of the recommended average saliencies of elements. We then simply define the saliency cue as

\[
D_s = |F - T|
\]  

(8)

Essentially, Eq. (8) suggests that if \( T(u) \) is very different from \( F(u) \), then the corresponding weight \( z_u \) is encouraged to be small by Eq. (1). Fig. 4 illustrates how similar elements from other images help to determine better elements.

**Foreground/Background Cue:** Another cue similar elements can provide is to recommend the given element to be foreground or background. For an element \( u \) and one of its similar elements \( v \), if their foreground feature descriptors are more similar than the background descriptors, \( v \) recommends foreground with a saliency punishment of \( (1 - T(u)) \) to \( u \); otherwise, it recommends background with a punishment of \( T(u) \), i.e.

\[
R_u(v) = |1 - T(u)|, \quad \text{if} \quad S_f(u, v) > S_b(u, v)
\]

\[
R_u(v) = |T(u) - 0|, \quad \text{if} \quad S_f(u, v) < S_b(u, v)
\]  

(9)

where \( R_u(v) \) denotes the saliency punishment recommended by \( v \) to \( u \). Considering all the similar elements, we define foreground/background cue \( D_F \) for an element \( u \) as

\[
D_F(u) = \frac{\sum_{v=1}^{N_n} \delta(S(u, v) > \theta) R_u(v)}{\sum_{v=1}^{N_n} \delta(S(u, v) > \theta)}
\]  

(10)

where \( \delta(\cdot) \) is the indication function, equal to one if the condition (\( \cdot \)) is true (otherwise 0), so as to include only similar elements. Fig. 4 also illustrates how similar elements from other images provide the foreground/background cue.

**Centerness Cue:** In addition to the above mentioned saliency and foreground/background cues, we also take advantage of the general observation that objects are often located at the center, and such central bias is quite prevalent in several benchmark datasets as pointed out in [35]. Therefore as an extra measure, saliency maps that emphasize center regions are encouraged to be given higher weights at central regions. To account for central bias, a spatial weight mask for each image is created using normalized Gaussian function which is centered at the image center. Specifically, for a pixel \( p \) in \( I^n \) (of size \( width_n \times height_n \)) with coordinates \((x, y)\) and with its origin at the image center, the central weight mask is defined as

\[
H^n(p) = exp\left(-\frac{x^2}{0.2 \times width^2_n} - \frac{y^2}{0.2 \times height^2_n}\right)
\]  

(11)

For an element \( u \) or \( e(n, k, m) \), its central bias is calculated by averaging the spatial weights of all its pixels, i.e.

\[
C_u = \frac{\sum_{p \in p^n_u} H^n(p)}{\sum_{p \in p^n_u} 1}
\]  

(12)

Let \( C = [C_1, C_2, \ldots, C_{N_n}] \) denote the vector consisting of the central bias weights of all the elements. Thus, we now define the centerness cue \( D_c \) for an element \( u \) as

\[
D_c(u) = C(u) \times |C(u) - T(u)|
\]  

(13)

which essentially measures how the saliency of an element deviates from its central bias weight. Central bias weight is also multiplied so that influence of this deviation in minimizing Eq. (1) depends upon the spatial location of the element.

Note that our centerness cue is different from other methods like [32], which deliberately emphasize the center regardless of whether an object is present or not in the center. On the contrary, our method emphasizes the center only if a salient object is present in the center. Our centerness cue provides additional support when the saliency and foreground/background cues fail to recommend something substantial because of lack of support from other images due to too much intra-class variation or pose differences. In such case, if there is a salient object at the center, it will be supported by the centerness cue.

**D. Smoothness Term**

Since in our prior term we have made discrete conditions using \( \theta \) to select similar elements, there is a certain possibility of inconsistencies in weight distribution. A smoothness term is necessary to curb inconsistencies in weight distribution among neighbor elements. Here we define neighbor elements as those which are similar in not only the feature space but also the saliency space. If a pair of elements have very similar saliency and are very close in the feature space as well, they should be encouraged to have similar weights. Thus, the smoothness term \( z'Gz \) is introduced to ensure that these neighbor elements in both feature space and saliency space take similar weights. However, we use the conventional normalized Laplacian matrix for defining smoothness term coefficient \( G \) in Eq. (1), similar to [36], i.e.

\[
G = A - Q^{-\frac{1}{2}}VQ^{-\frac{1}{2}}
\]  

(14)

where \( A \) is the identity matrix, \( V \) is neighborhood matrix, and \( Q \) is the diagonal matrix composed of row sums of matrix \( V \). In addition, different from the similarity matrix \( S \) defined in Eq. (5), \( V \) takes into account similarity in both feature space and saliency space, i.e.

\[
V(u, v) = exp\left(-\gamma \frac{\sum_{q=1}^{2d} (X(u,q) - X(v,q))^2}{2d} - |T(u) - T(v)|\right)
\]  

(15)

where \( \gamma \) is a normalization parameter set to \( \frac{1}{300} \), which is the same as that in Eqs. (3), (4) and (5).

**E. Implementation Details**

For optimization, since \( G \) is positive semi-definite and the constraints are linear, the objective function defined in (1) is essentially a quadratic programming problem, which is solved by the interior-point convex algorithm provided in Matlab.

Once the fused saliency map is available, different single-image segmentation algorithms can be applied for segmentation. In this research, we adopt two segmentation methods as
two variations. One is the classical Otsu’s method, which is an optimal threshold based method. The other one is GrabCut algorithm [37] with some modification. Specifically, by noticing the final fused saliency map containing certain boundary information, following [17], we modify the GrabCut energy equation and add another localization potential to ensure that segmentation is guided not only by color, but also by the location prescribed by the object prior contained in the fused saliency map. The foreground (FG) and the background (BG) seed locations are determined by

$$p \in \begin{cases} 
  FG, & \text{if } J^n(p) > \tau \\
  BG, & \text{if } J^n(p) < \phi^n
\end{cases}$$

(16)

where $\phi^n$ is a global threshold value automatically determined by the classical Otsu method and $\tau$ is a parameter (by default $\tau = 0.75$). It should be noted that other single-image segmentation methods such as [38] can also be used for the final segmentation.

IV. EXPERIMENTAL RESULTS

We conducted extensive experiments on five existing benchmark co-segmentation datasets (MSRC [47], iCoseg [8], Coseg-Rep [19], Internet image dataset [16], and FlickrMFC dataset [14]). As mentioned in the introduction, the existing methods often require fine tuning of quite a few parameters. In order to demonstrate the effectiveness of our method, we make two types of settings in our experiments: (1) default parameter settings for all the categories in the datasets and (2) tuning parameter $\tau$ over categories for a fair comparison with other methods. Following the literature, we adopted two evaluation metrics: (i) Jaccard Similarity (Jacc.) [48] and (ii) Accuracy (Acc.). Denote $A^F_p$, $A^B_p$, $A^F_y$, and $A^B_y$ as proposed foreground pixels set, proposed background pixels set, groundtruth foreground pixels set and groundtruth background pixels set, respectively. Here, Jaccard Similarity is defined as the size of intersection divided by the size of union of the proposed and groundtruth foreground pixels sets, i.e., $\frac{|A^F_p \cap A^F_y|}{|A^F_p \cup A^F_y|}$. And Accuracy is defined as the percentage of pixels that have same labels in both the proposed and groundtruth masks, i.e., $\frac{|A^F_p \cap A^F_y| + |A^B_p \cap A^B_y|}{|A^F_p \cup A^F_y|} \times 100$. We use eight saliency extraction methods [39][40][41][42][43][44][45][46] to generate various saliency maps as the input to our method. In the following subsections, we first briefly introduce the datasets used, followed by individual experiments, discussions and comparisons.

A. Datasets

MSRC dataset contains 14 categories with 418 images in total. Coseg-Rep dataset contains 23 categories and 572 images in total, where there is a special category named “Repetitive” that has several instances of the same type of object within one image (e.g., an image containing multiple horses). Both of the datasets exhibit intra-class variation. As a result, we do not use the color feature for matching the elements as it will be unreliable.

iCoseg dataset contains 38 categories with 643 images in total. For a fair comparison with the existing methods [16], [20], we use the same part of the dataset in our experiments. This includes 30 categories and a total of 530 images. FlickrMFC dataset contains multiple common objects that might not appear in every image. It has 14 categories and 263 images in total. For these two datasets, since the same objects appear frequently across the images, we include the color features in our method. Also, Internet images dataset created by [16] contains 3 categories: Airplane, Car, and Horse, with 4347, 418, and 572 images in total, respectively.
6381 and 4542 images respectively, where only some of the images have ground-truth. Again, due to intra-class variation, we avoid using color features for this dataset.

Note that we first perform k-means clustering using GIST descriptor [29] and the proposed saliency co-fusion is then applied to each cluster independently. This is to reduce the intra-class variation. Otherwise, a wide diversity might cause unnecessary difficulties in the co-fusion process. Empirically, we set the target cluster size to be 10, i.e. on average each cluster contains 10 images.

### B. Performance Improvement by Co-fusion

The key point of our proposed saliency co-fusion process is to generate a fused saliency map that can better highlight the common object while suppressing the background saliency. To compare the quality of the fused saliency map with other saliency maps, we apply the simple segmentation approach, Otsu’s method, on individual saliency maps of images in MSRC dataset, and report segmentation results in Table II. It can be seen that our method achieves about 10% gain over that of the best saliency extraction method [46]. Table II also shows the results of simple averaging or taking the maximum of those individual saliency maps at the pixel level also outperform the best single saliency map, clearly suggesting the advantage of using multiple saliency maps. Our method outperforms the simple average function and the max function by about 6% and 8%, respectively. Note that the Avg Jaccard Similarity value of 0.688 on MSRC dataset by using simple Otsu’s method is even better than the result of 0.68 (see Table VII) obtained by [16] which used complex co-labelling, parameter tuning, and Grabcut.

![Fig. 5. Examples to illustrate the advantages of the fused saliency maps over the input saliency maps.](image-url)

Table II

<table>
<thead>
<tr>
<th>class</th>
<th>[39]</th>
<th>[40]</th>
<th>[41]</th>
<th>[42]</th>
<th>[43]</th>
<th>[44]</th>
<th>[45]</th>
<th>[46]</th>
<th>AVG</th>
<th>MAX</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Car</td>
<td>0.541</td>
<td>0.466</td>
<td>0.381</td>
<td>0.469</td>
<td>0.510</td>
<td>0.598</td>
<td>0.507</td>
<td>0.629</td>
<td>0.660</td>
<td>0.666</td>
<td>0.696</td>
</tr>
<tr>
<td>Sheep</td>
<td>0.745</td>
<td>0.699</td>
<td>0.736</td>
<td>0.612</td>
<td>0.776</td>
<td>0.615</td>
<td>0.697</td>
<td>0.744</td>
<td>0.793</td>
<td>0.779</td>
<td>0.810</td>
</tr>
<tr>
<td>Cow</td>
<td>0.670</td>
<td>0.670</td>
<td>0.673</td>
<td>0.603</td>
<td>0.734</td>
<td>0.658</td>
<td>0.653</td>
<td>0.736</td>
<td>0.742</td>
<td>0.729</td>
<td>0.794</td>
</tr>
<tr>
<td>Flower</td>
<td>0.705</td>
<td>0.679</td>
<td>0.536</td>
<td>0.627</td>
<td>0.694</td>
<td>0.625</td>
<td>0.641</td>
<td>0.688</td>
<td>0.721</td>
<td>0.726</td>
<td>0.768</td>
</tr>
<tr>
<td>Cat</td>
<td>0.439</td>
<td>0.526</td>
<td>0.560</td>
<td>0.597</td>
<td>0.573</td>
<td>0.565</td>
<td>0.539</td>
<td>0.609</td>
<td>0.651</td>
<td>0.624</td>
<td>0.714</td>
</tr>
<tr>
<td>Sign</td>
<td>0.746</td>
<td>0.619</td>
<td>0.552</td>
<td>0.567</td>
<td>0.646</td>
<td>0.669</td>
<td>0.570</td>
<td>0.796</td>
<td>0.775</td>
<td>0.743</td>
<td>0.812</td>
</tr>
<tr>
<td>Tree</td>
<td>0.636</td>
<td>0.655</td>
<td>0.471</td>
<td>0.400</td>
<td>0.632</td>
<td>0.601</td>
<td>0.561</td>
<td>0.606</td>
<td>0.681</td>
<td>0.669</td>
<td>0.738</td>
</tr>
<tr>
<td>House</td>
<td>0.586</td>
<td>0.613</td>
<td>0.486</td>
<td>0.389</td>
<td>0.528</td>
<td>0.630</td>
<td>0.450</td>
<td>0.640</td>
<td>0.670</td>
<td>0.669</td>
<td>0.712</td>
</tr>
<tr>
<td>Dog</td>
<td>0.527</td>
<td>0.559</td>
<td>0.503</td>
<td>0.520</td>
<td>0.469</td>
<td>0.452</td>
<td>0.503</td>
<td>0.627</td>
<td>0.628</td>
<td>0.582</td>
<td>0.643</td>
</tr>
<tr>
<td>Bird</td>
<td>0.529</td>
<td>0.590</td>
<td>0.573</td>
<td>0.535</td>
<td>0.590</td>
<td>0.459</td>
<td>0.583</td>
<td>0.611</td>
<td>0.644</td>
<td>0.589</td>
<td>0.662</td>
</tr>
<tr>
<td>Bike</td>
<td>0.377</td>
<td>0.416</td>
<td>0.297</td>
<td>0.3827</td>
<td>0.436</td>
<td>0.453</td>
<td>0.463</td>
<td>0.420</td>
<td>0.488</td>
<td>0.473</td>
<td>0.548</td>
</tr>
<tr>
<td>Chair</td>
<td>0.546</td>
<td>0.588</td>
<td>0.566</td>
<td>0.474</td>
<td>0.595</td>
<td>0.530</td>
<td>0.496</td>
<td>0.563</td>
<td>0.638</td>
<td>0.588</td>
<td>0.638</td>
</tr>
<tr>
<td>Face</td>
<td>0.515</td>
<td>0.411</td>
<td>0.395</td>
<td>0.582</td>
<td>0.463</td>
<td>0.446</td>
<td>0.367</td>
<td>0.548</td>
<td>0.565</td>
<td>0.567</td>
<td>0.571</td>
</tr>
<tr>
<td>Plane</td>
<td>0.420</td>
<td>0.437</td>
<td>0.399</td>
<td>0.297</td>
<td>0.505</td>
<td>0.505</td>
<td>0.475</td>
<td>0.542</td>
<td>0.535</td>
<td>0.469</td>
<td>0.518</td>
</tr>
<tr>
<td>Avg</td>
<td>0.570</td>
<td>0.566</td>
<td>0.510</td>
<td>0.504</td>
<td>0.582</td>
<td>0.558</td>
<td>0.536</td>
<td>0.626</td>
<td>0.656</td>
<td>0.634</td>
<td>0.688</td>
</tr>
</tbody>
</table>

Fig. 5 shows the visual comparison of individual saliency maps used and our fused saliency map. It can be seen that pixels pertaining to the woman (the common object) obtain boosted saliency values, while the background regions get suppressed saliency values in the final fused saliency maps which lead to clean segmentation results. Fig. 6 provides more examples of fused saliency maps and the corresponding segmentation results on MSRC and iCoseg datasets. Furthermore, it also shows the difference maps against the ground-truths.
C. Discussion on the Parameters

In Table III, we report our results obtained by fixing the parameter $\tau$ in Eq. (16) to 0.75 on all the datasets with GrabCut segmentation, and also the results obtained using simple Otsu’s method. Due to the fact that categories of Internet images dataset are quite large, their results on each category are separately shown. We can see that even the simple Otsu’s method is able to produce decent results with our fused saliency maps. This can be attributed to the high-quality saliency maps produced by our saliency co-fusion approach. By using GrabCut for segmentation, the performance of our method can be further improved. For parameter $\lambda$ in Eq. (1), we empirically set it to 9. Also, we empirically set parameter $\gamma$ in Eqs. (3), (4), (5), and (15) to 1/300, and parameter $\theta$ in Eqs. (7) and (10) to 0.75. Parameter $\phi^n$ in Eq. (16) is automatically computed using Ostu’s method.

In order to examine the sensitivity of our method on different superpixel extraction methods and different parameter settings, we further conducted experiments using irregular

---

**TABLE III**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Otsu’s method</th>
<th>GrabCut</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSRC</td>
<td>0.69</td>
<td>86.7</td>
</tr>
<tr>
<td>iCoseg</td>
<td>0.65</td>
<td>87.0</td>
</tr>
<tr>
<td>Coseg-Rep</td>
<td>0.71</td>
<td>89.5</td>
</tr>
<tr>
<td>Car</td>
<td>0.70</td>
<td>85.3</td>
</tr>
<tr>
<td>Horse</td>
<td>0.49</td>
<td>78.5</td>
</tr>
<tr>
<td>Airplane</td>
<td>0.52</td>
<td>82.6</td>
</tr>
<tr>
<td>FlickrMFC</td>
<td>0.60</td>
<td>83.5</td>
</tr>
</tbody>
</table>

---

**TABLE IV**

Performance results by varying region-size parameter of SLIC [34] on MSRC dataset

<table>
<thead>
<tr>
<th>Region Size</th>
<th>Jacc.</th>
<th>Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>0.6878</td>
<td>86.31</td>
</tr>
<tr>
<td>40</td>
<td>0.6877</td>
<td>86.30</td>
</tr>
<tr>
<td>60</td>
<td>0.6870</td>
<td>86.27</td>
</tr>
<tr>
<td>80</td>
<td>0.6869</td>
<td>86.26</td>
</tr>
<tr>
<td>100</td>
<td>0.6865</td>
<td>86.25</td>
</tr>
</tbody>
</table>
superpixels generated by [49]. The results on MSRC dataset show that use of the superpixels of [49] with the global thresholding achieves the average Jaccard similarity of 0.6876. However, this is almost same as the result of 0.6875 obtained by using SLIC [34]. We also vary the region-size parameter of SLIC [34]. By varying the region-size parameter of SLIC [34] from 20 to 100, the results can be seen in Table IV. It can be seen that the performance decreases only slightly with the increase of the region size. Therefore, these experiments indicate that the proposed method is robust to different superpixel methods/settings.

D. Experiments for Comparison

For different datasets, we compare our method with the methods that report the state-of-the-art performance on the datasets. We denote “Ours (default)” as our method with the setting $\tau = 0.75$ using GrabCut while denoting “Ours (tuned)” as the one where we tune parameter $\tau$ with a step size of 0.03 from 0.60 to 0.99 over each category and report the best results, which is similar to other methods. Our method outperforms the state-of-the-art methods on two of the single object co-segmentation datasets (Coseg-Rep and Internet Images) as shown in Tables V and VI. Also, some sample visual results of our method on Coseg-Rep dataset and Internet images dataset are shown in Fig. 7 and Fig. 8, respectively.

Note that for the Internet image dataset, since each of its categories consists of large number of images, we tune parameter $\tau$ per cluster. It can be seen from Tables V and VI that, in terms of Jaccard Similarity metric, our method achieves about 5% on Coseg-Rep dataset, 13%, 11%, and 9% improvements on
Fig. 9. Sample segmentation results on FlickrMFC dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>MSRC Jacc.</th>
<th>MSRC Acc.</th>
<th>iCoseg Jacc.</th>
<th>iCoseg Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discriminative [4]</td>
<td>0.45</td>
<td>70.8</td>
<td>0.39</td>
<td>61.0</td>
</tr>
<tr>
<td>Multi-Class [50]</td>
<td>0.51</td>
<td>73.6</td>
<td>0.43</td>
<td>70.2</td>
</tr>
<tr>
<td>Object Discovery [16]</td>
<td>0.68</td>
<td>87.7</td>
<td>0.69</td>
<td>89.8</td>
</tr>
<tr>
<td>Geometric Mean Saliency [33]</td>
<td>0.70</td>
<td>88.4</td>
<td>0.72</td>
<td>91.6</td>
</tr>
<tr>
<td>Composition [20]</td>
<td>0.73</td>
<td>89.2</td>
<td>0.73</td>
<td>92.8</td>
</tr>
<tr>
<td>Ours (tuned)</td>
<td>0.71</td>
<td>88.7</td>
<td>0.72</td>
<td>91.9</td>
</tr>
</tbody>
</table>

It is interesting to see that our method can also well handle Flickr MFC dataset that contains multiple common objects across the images and the repetitive category of Coseg-Rep dataset that contains repeated instances of objects, as shown in Tables VIII and XII, respectively. Our method with tuning per category outperformed the best one [12] (with supervised information) by 6% in terms of Jaccard similarity metric despite being an unsupervised method. In fact, our method’s default setting itself outperforms the state-of-the-art method on Flickr MFC dataset. It should be noted that the comparison here is in terms of foreground/background segregation, and not multi-label segmentation. Fig. 9 shows some sample segmentation results in such multiple-foreground scenario. It can be seen that although different multiple objects are present in one category Car, Horse and Airplane categories of Internet image dataset, respectively, when compared with the best results reported in [33] and [16] for CosegRep and Internet Images datasets, respectively. Table VII compares the results of our method with those of state-of-the-art methods on MSRC and iCoseg datasets. It can be seen that our results are competitive to the best one by [20], while our method is much faster than [20]. Specifically, running on the same PC with Intel Core i5-3470@3.20 GHz CPU and 32 GB RAM, [20] (using their own source codes in Matlab) takes 29.2 hours to complete the entire segmentation process on MSRC dataset. However, our method (also in Matlab codes) takes only 8.5 hours. These durations include the time taken for pre-processing steps as well like generating proposals in [20] and generating saliency maps in our method.

Table VIII

Comparison on FlickrMFC dataset using overall Jaccard Similarity (Jacc.) value. (U) means unsupervised method and (S) means supervised method

<table>
<thead>
<tr>
<th>Methods</th>
<th>Jacc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple Foreground Cosegmentation (U) [14]</td>
<td>0.322</td>
</tr>
<tr>
<td>Multiple Foreground Cosegmentation (S) [14]</td>
<td>0.482</td>
</tr>
<tr>
<td>Discriminative Clustering (U) [50]</td>
<td>0.414</td>
</tr>
<tr>
<td>Directed Graph Clustering (U) [11]</td>
<td>0.547</td>
</tr>
<tr>
<td>Graph Transduction (S) [13]</td>
<td>0.626</td>
</tr>
<tr>
<td>w/o NON RIGID Mapping (U) [12]</td>
<td>0.589</td>
</tr>
<tr>
<td>with NON RIGID Mapping (S) [12]</td>
<td>0.647</td>
</tr>
<tr>
<td>Ours (U) (default)</td>
<td>0.667</td>
</tr>
<tr>
<td>Ours (U) (tuned)</td>
<td>0.684</td>
</tr>
</tbody>
</table>

Car, Horse and Airplane categories of Internet image dataset, respectively, when compared with the best results reported in [33] and [16] for CosegRep and Internet Images datasets, respectively. Table VII compares the results of our method with those of state-of-the-art methods on MSRC and iCoseg datasets. It can be seen that our results are competitive to the best one by [20], while our method is much faster than [20]. Specifically, running on the same PC with Intel Core i5-3470@3.20 GHz CPU and 32 GB RAM, [20] (using their own source codes in Matlab) takes 29.2 hours to complete the entire segmentation process on MSRC dataset. However, our method (also in Matlab codes) takes only 8.5 hours. These durations include the time taken for pre-processing steps as well like generating proposals in [20] and generating saliency maps in our method.

It is interesting to see that our method can also well handle Flickr MFC dataset that contains multiple common objects across the images and the repetitive category of Coseg-Rep dataset that contains repeated instances of objects, as shown in Tables VIII and XII, respectively. Our method with tuning per category outperformed the best one [12] (with supervised information) by 6% in terms of Jaccard similarity metric despite being an unsupervised method. In fact, our method’s default setting itself outperforms the state-of-the-art method on Flickr MFC dataset. It should be noted that the comparison here is in terms of foreground/background segregation, and not multi-label segmentation. Fig. 9 shows some sample segmentation results in such multiple-foreground scenario. It can be seen that although different multiple objects are present in one category Car, Horse and Airplane categories of Internet image dataset, respectively, when compared with the best results reported in [33] and [16] for CosegRep and Internet Images datasets, respectively. Table VII compares the results of our method with those of state-of-the-art methods on MSRC and iCoseg datasets. It can be seen that our results are competitive to the best one by [20], while our method is much faster than [20]. Specifically, running on the same PC with Intel Core i5-3470@3.20 GHz CPU and 32 GB RAM, [20] (using their own source codes in Matlab) takes 29.2 hours to complete the entire segmentation process on MSRC dataset. However, our method (also in Matlab codes) takes only 8.5 hours. These durations include the time taken for pre-processing steps as well like generating proposals in [20] and generating saliency maps in our method.

It is interesting to see that our method can also well handle Flickr MFC dataset that contains multiple common objects across the images and the repetitive category of Coseg-Rep dataset that contains repeated instances of objects, as shown in Tables VIII and XII, respectively. Our method with tuning per category outperformed the best one [12] (with supervised information) by 6% in terms of Jaccard similarity metric despite being an unsupervised method. In fact, our method’s default setting itself outperforms the state-of-the-art method on Flickr MFC dataset. It should be noted that the comparison here is in terms of foreground/background segregation, and not multi-label segmentation. Fig. 9 shows some sample segmentation results in such multiple-foreground scenario. It can be seen that although different multiple objects are present in one category Car, Horse and Airplane categories of Internet image dataset, respectively, when compared with the best results reported in [33] and [16] for CosegRep and Internet Images datasets, respectively. Table VII compares the results of our method with those of state-of-the-art methods on MSRC and iCoseg datasets. It can be seen that our results are competitive to the best one by [20], while our method is much faster than [20]. Specifically, running on the same PC with Intel Core i5-3470@3.20 GHz CPU and 32 GB RAM, [20] (using their own source codes in Matlab) takes 29.2 hours to complete the entire segmentation process on MSRC dataset. However, our method (also in Matlab codes) takes only 8.5 hours. These durations include the time taken for pre-processing steps as well like generating proposals in [20] and generating saliency maps in our method.

It is interesting to see that our method can also well handle Flickr MFC dataset that contains multiple common objects across the images and the repetitive category of Coseg-Rep dataset that contains repeated instances of objects, as shown in Tables VIII and XII, respectively. Our method with tuning per category outperformed the best one [12] (with supervised information) by 6% in terms of Jaccard similarity metric despite being an unsupervised method. In fact, our method’s default setting itself outperforms the state-of-the-art method on Flickr MFC dataset. It should be noted that the comparison here is in terms of foreground/background segregation, and not multi-label segmentation. Fig. 9 shows some sample segmentation results in such multiple-foreground scenario. It can be seen that although different multiple objects are present in one category
of the dataset, our method successfully extracts the foreground. As far as the repetitive scenario is concerned, our method obtained a Jaccard Similarity value of 0.776 in comparison to 0.754 obtained by [19] on the repetitive category of the Coseg-Rep dataset (see bottom three rows of Fig. 7 for such sample visual results).

Tables IX-XII list out the detailed Jaccard similarity results of our method as well as the state-of-the-art methods on individual categories of the four datasets. It was seen earlier that our method performs worse than [20] on MSRC and iCoseg datasets as far as the overall average performance is concerned. The main reason could be that our method relies on saliency co-fusion. If the common object cannot be identified as salient by any of the saliency extraction methods, our method would not be able to segment it out. Interestingly, these tables reveal that despite such slightly inferior overall performance, our method outperforms [20] in 4 out of 14 and 15 out of 30 categories in MSRC and iCoseg datasets, respectively. Fig. 10 gives some visual examples of those categories, where our results look better than those of [20].

![Fig. 10. Sample segmentation results where our method outperforms [20].](image-url)
V. Conclusion

We have proposed a novel saliency co-fusion approach for the purpose of image co-segmentation which uses the association of similar images to fuse multiple saliency maps of an image in order to boost up common foreground saliency and suppress background saliency. Experimental results on five benchmark datasets show that our method while co-fusing eight different saliency maps, achieves very competitive performance, compared to the state-of-the-art methods of image co-segmentation.

Acknowledgment

This research was carried out at the Rapid-Rich Object Search (ROSE) Lab at the Nanyang technological University, Singapore. The ROSE Lab is supported by the National Research Foundation, Prime Minister’s Office, Singapore, under its IDM Futures Funding Initiative and administered by the Interactive and Digital Media Programme Office. This research is also partially supported by Singapore MoE AcRF Tier-1 Grant RG138/14.

References


E. Limitations and Discussions

Although our method performs well on the benchmark datasets in general, there are some failure cases: (i) As shown in the red-box of Fig. 11, our method only segments out one dog and misses the other. This is because one of the dogs is extremely salient in all the saliency maps, while the other dog is not very salient in any of the saliency maps. (ii) Another case is when there are multiple common salient objects in the images, while the goal of benchmark dataset is to segment out only one common object. For such case, our method will segment out all the salient common objects as shown in the green-box of Fig. 11. (iii) Similar to almost all the co-segmentation methods, our method requires sufficient background variations across the images in one cluster. If very similar images are being included in one cluster, our method will fail to distinguish background from the foreground, as illustrated in the yellow-box of Fig. 11.

The blue box in Fig. 11 gives some class-specific examples where our method does not perform well. For example, (a) Bicycles in the bike category need segmentation of thin rods and tyres whereas our method segments such bicycles into convex shapes such as triangles and disks due to using GrabCut; (b) Our method misses segmenting out shoulders in most of the images in the face category, because shoulders are not so salient; and (c) Many images in the plane category also include airports along with the planes, thus making it difficult to segment out the planes clearly.

REFERENCES


Koteswar Rao Jerripothula (S’15) received his B.Tech degree in Electrical Engineering from Indian Institute of Technology Roorkee, India in 2012, and is currently working towards his PhD degree at Nanyang Technological University, Singapore. His research interests include computer vision and multimedia applications.

Jianfei Cai (S’08–M’02–SM’07) received his PhD degree from the University of Missouri-Columbia. He is currently an Associate Professor and has served as the Head of Visual & Interactive Computing Division and the Head of Computer Communication Division at the School of Computer Engineering, Nanyang Technological University, Singapore. His major research interests include computer vision, visual computing and multimedia networking. He has published more than 170 technical papers in international journals and conferences. He has been actively participating in program committees of various conferences. He has served as the leading Technical Program Chair for IEEE International Conference on Multimedia & Expo (ICME) 2012 and the leading General Chair for Pacific Rim Conference on Multimedia (PCM) 2012. Since 2013, he has been serving as an Associate Editor for IEEE Trans on Image Processing (T-IP). He has also served as an Associate Editor for IEEE Trans on Circuits and Systems for Video Technology (T-CSVT) from 2006 to 2013.

Transactions on Multimedia
Junsong Yuan (M’08–SM’14) is associate professor and program director of video analytics at School of Electrical and Electronics Engineering (EEE), Nanyang Technological University (NTU). He received Ph.D. from Northwestern University in 2009 and M.Eng. from National University of Singapore in 2005. Before that, he graduated from the Special Class for the Gifted Young of Huazhong University of Science and Technology in 2002. His research interests include computer vision, video analytics, gesture and action analysis, large-scale visual search and mining, etc. He has published 3 books, 5 book chapters, 150 conference and journal papers, and filed several patents, with technology licensed by the industry. He is Program Co-Chair of IEEE Conf. on Visual Communications and Image Processing (VCIP’15), Organizing Co-Chair of Asian Conf. on Computer Vision (ACCV’14), Area Chair of ACCV’14, WACV’14, ICME’14’15, and co-chairs 6 workshops at CVPR/ICCV/SIGGRAPH Asia. He serves as guest editor of International Journal of Computer Vision (IJCV), and is currently associate editor of IEEE Trans. on Image Processing (T-IP), IEEE Trans. on Circuits and Systems for Video Technology (T-CSVT) and The Visual Computer journal (TVC).

He received Nanyang Assistant Professorship and Tan Chin Tuan Exchange Fellowship from Nanyang Technological University, Outstanding EECS Ph.D. Thesis award from Northwestern University, Doctoral Spotlight Award from IEEE CVPR’09, and National Outstanding Student from Ministry of Education, P.R.China.