Co-Segmentation by Composition

Alon Faktor* Michal Irani
Dept. of Computer Science and Applied Math
The Weizmann Institute of Science,ISRAEL

Abstract

Given a set of images which share an object from the same semantic category, we would like to co-segment the shared object. We define 'good' co-segments to be ones which can be easily composed (like a puzzle) from large pieces of other co-segments, yet are difficult to compose from remaining image parts. These pieces must not only match well but also be statistically significant (hard to compose at random). This gives rise to co-segmentation of objects in very challenging scenarios with large variations in appearance, shape and large amounts of clutter. We further show how multiple images can collaborate and "score" each others' co-segments to improve the overall fidelity and accuracy of the co-segmentation. Our co-segmentation can be applied both to large image collections, as well as to very few images (where there is too little data for unsupervised learning). At the extreme, it can be applied even to a single image, to extract its co-occurring objects. Our approach obtains state-of-the-art results on benchmark datasets. We further show very encouraging co-segmentation results on the challenging PASCAL-VOC dataset.

1. Introduction

As the amount of visual and semantic information in the web grows, there is an increasing need for methods to organize and mine it. These methods should automatically extract additional semantic information from the weak available information. For example, from the existing tagged images on the web, one would like to obtain more accurate information such as localization or even segmentation of the main objects in each image.

In this work, we focus on the "object co-segmentation" problem - given a set of images which share an object from the same semantic category, we would like to co-segment the common objects. Existing work in this field has typically assumed a simple model common to the co-objects such as common color [16, 13] or common distribution of descriptors [19]. More sophisticated models for co-segmentation were proposed, such as a discriminative model between the foreground and the background [11, 7] or generative models of the co-objects [1, 22]. In [21], a similarity measure between segments was learned for mining the co-objects from an initial pool of object proposals.

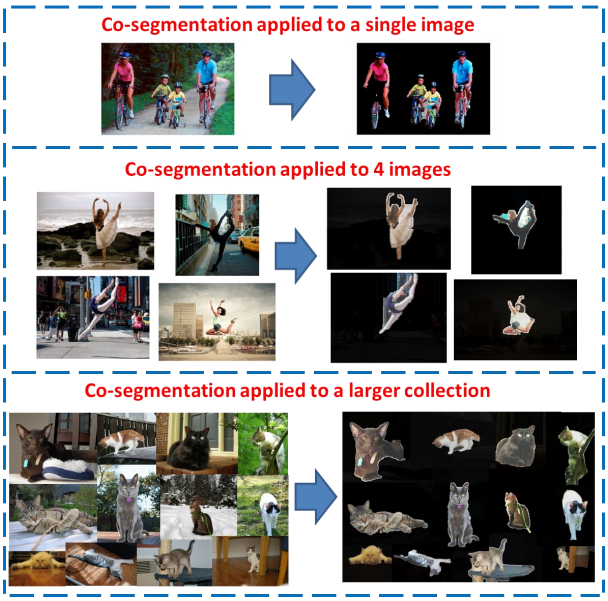

Figure 1. **Co-Segmentations produced by our algorithm.**

However, observing the co-occuring objects in Fig. 1 (bike riders, ballet dancers, cats), we can see that there seems to be no simple model common to the objects. The co-objects within each set, may have different colors, poses and shapes. Moreover, the objects may not be salient in their image and may be surrounded by large amounts of distracting clutter. These kinds of dataset form a challenge to existing methods.

In this paper we suggest an approach for co-segmentation, which does not rely on any simple model common to the co-objects. Instead, our approach is based on the framework developed in [9, 5], which show that when non-trivial (rare) image parts re-occur in another image, they induce *statistically meaningful affinities* between these images. However, unlike [9] which employs this idea to induce affinities between *entire* images (for the purpose of image clustering), we employ their approach to induce

*Funded in part by the Israeli Science Foundation and the Israeli Ministry of Science.

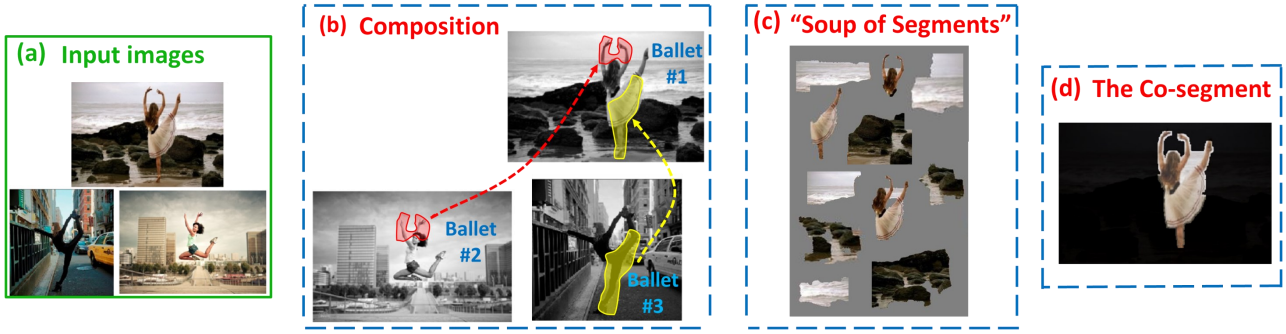

Figure 2. **"Co-Segmentation by composition".** (a) The input images. (b) Co-occurring regions induce affinities between image parts across images. (c) "Soup of Segments". (d) The final co-segment.

affinities between *parts* of images, thus initializing our co-segmentation process.

We expect "good" co-segments to be, on the one hand, good image segments, and on the other hand, to be well composed from other co-segments. That is, a co-segment should share large non-trivial (statistically significant) regions with other co-segments. Yet, it should not be easily composable from image parts outside the co-segments.

Our approach is composed of three main building blocks:

I. **Initialize the co-segmentation by inducing affinities between image parts** - Large shared regions are detected across images, inducing affinities between those image parts (see Fig. 2b). The larger and more rare those regions are, the higher their induced affinity. The shared regions provide a *rough* localization of the co-objcets in the images. The region detection is done efficiently using a randomized search and propagation algorithm, suggested by [9]. These ideas are detailed in Sec. 3.

II. **From co-occurring regions to co-segments** - The detected shared regions are usually not good image segments on their own. They are not confined to image edges, and may cover only part of the co-objects. However, they induce statistically significant affinities between parts of co-objects. We use these affinities to score multiple overlapping segment candidates ("soup of segments" – see Fig. 2c). A segment which is highly overlapped by many shared regions gets a high score. The segments and their scores are then used to estimate co-segmentation likelihood maps (See Fig. 3b). These ideas are detailed in Sec. 4.1.

III. **Improving co-segmentation by "consensus scoring" -** We improve the the fidelity and accuracy of the co-segmentation by propagating the co-segmentation likelihood maps between the different images. This propagation is done using the mapping between the co-occurring (shared) regions across the different images. The co-segmentation score is determined using the *consensus* between each region and its co-occurring regions in other images. This leads to improved co-segmentation likelihood maps (see Fig. 3c). These ideas are detailed in Sec. 4.2.

Finally, we show results of applying our co-segmentation both to large image collections, as well as to very few images (where there is too little data for unsupervised learning). We further show that it can even be applied to a *single* image, to extract its co-occurring objects. Some such examples are shown in Fig. 1. Our approach obtains state-of-the-art results on benchmark datasets. We further show very encouraging co-segmentation results on the challenging PASCAL-VOC dataset. These are detailed in Secs. 5, 6.

2. Closely Related Work

Co-segmentation methods which employ region correspondence were also suggested by [18] and [12]. However, their regions are image segments which are extracted from each image separately ahead of time and then matched. In contrast, our shared regions are usually not good image segments that can be extracted ahead of time. What makes them "good" image regions is the fact that (i) they are rare (have low chance of occurring at random), yet (ii) they co-occur (are shared) by two images. When such a *rare* region co-occurs, it is unlikely to be accidental, thus inducing high meaningful affinity between those image parts.

Recently, [17] suggested to combine visual saliency and dense pixel correspondences across images for the purpose of co-segmentation. We also employ dense correspondences for detecting our shared regions. However, we use the statistical significance of the shared regions to initialize the co-segmentation and not visual saliency like [17] does. This enables us to perform co-segmentation, even if many of the co-objects are are not salient within their images.

[14] suggested to incorporate into co-segmentation generic knowledge transfer from datasets with human-annotated segmentations of objects. We also transfer knowledge from "soft" segmentation maps of other images using our "consensus scoring". However, we do not rely on any external human-annotated dataset, but use only the images we wish to co-segment. Despite this, we are able to obtain results comparable to [14], as will be shown Sec. 6.

3. Inducing Affinities between Image Parts

Our framework for inducing affinities between image parts is based on [5] and [9]. To make the paper self-contained, we briefly review their main ideas.

3.1. "Similarity by Composition"

In the framework of [5], one image is inferred as being similar to another image if it can be easily composed (like a puzzle) from a few large pieces of the second image. These regions must both match well, as well as be statistically significant (hard to compose at random).

The co-occurring regions induce affinities between image parts across different images. We use the definition of [5] for computing the affinity of a co-occurring region R between two images I_1, I_2:

$$\text{Aff}(R|I_1, I_2) = \log \frac{p(R|I_1, I2)}{p(R|H_0)} \quad (1)$$

where $p(R|I_1, I2)$ measures the degree of similarity of the regions found in these two images and $p(R|H_0)$ measures the chance of the region to occur at random (be generated by a random process H_0). If a region matches well, but is trivial, then its likelihood ratio will be low (inducing a low affinity). On the other hand, if a region is non-trivial, yet has a good match in another image, its likelihood ratio will be high (inducing a high affinity).

The following approximations were made by [9] in order to get a simple expression for Eq(1):

1. Represent the region $R \subset I_1$ by densely sampled descriptors $\{d_i\}$. Assume $\{d_i\}$ are i.i.d. The likelihood of each descriptor d_i in the region $R \subset I_1$ to be generated from I_2 is approximated by:

$$p(d_i|I_1, I_2) = \exp\left(-\left|\Delta d_i(I_1, I_2)\right|^2 / 2\sigma^2\right) \quad (2)$$

where $\Delta d_i(I_1, I_2)$ is the matching error of d_i (i.e, the l_2 distance between d_i and its corresponding descriptor in its region match in the other image I_2).

2. Approximate the random process H_0 by generating a descriptor codebook $\hat{D}$ (with a few hundred codewords). This codebook is generated by applying k-means clustering to all of the descriptors extracted from the image collection. Frequent descriptors will be represented well in $\hat{D}$ (have low error relative to their nearest codeword). Rare descriptors, on the other hand, will be represented poorly in $\hat{D}$ (have high error relative to their nearest codeword).Thus, the likelihood of each descriptor d_i in the region $R \subset I_1$ to be generated at random (using H_0) is approximated by:

$$p(d_i|I_1, I_2) = \exp\left(-\left|\Delta d_i(H_0)\right|^2 / 2\sigma^2\right) \quad (3)$$

where $\Delta d_i(H_0)$ is the error of descriptor d_i with respect to the codebook (i.e, the l_2 distance between d_i and its nearest neighbor descriptor in the codebook $\hat{D}$)[1].

This yields the following expression for $\text{Aff}(R|I_1, I_2)$:

$$\text{Aff}(R|I_1, I_2) = \sum_{d_i \in R} \left|\Delta d_i(H_0)\right|^2 - \left|\Delta d_i(I_1, I_2)\right|^2 \quad (4)$$

Namely, the affinity induced by a co-occurring region is equal to the difference between the total descriptor error with respect to a codebook and the total matching error between the matched regions in the two images. A high affinity will be obtained for image parts which are both rare (high codebook errors) and match well across images (low matching errors). These image parts tend to coincide with unique and informative parts of the co-occurring objects, yielding a good seed to the co-segments.

The notion of composition is illustrated in Fig 2. Ballet dancer #1 appears in a different pose than any of the two other Ballet dancers. However, Ballet dancer #1 can compose its arm gesture (red region) from Ballet dancer #2 and most of its leg gesture (yellow region) from Ballet dancer #3. Note that these regions are complex, thus have a low chance of appearing at random. Therefore, the fact that these regions found good matches in other images can not be accidental, providing high evidence to the high affinity between those regions.

3.2. Detecting Co-occurring Regions between Images

Detecting large non-trivial co-occurring regions between images is in principle a very hard problem (already between a pair of images, let alone in a large image collection). Moreover, the regions may be of *arbitrary size and shape*. Therefore, [9] suggested a randomized search algorithm which guarantees with *very high probability* the efficient detection of large shared regions. Regions are represented by densely sampled descriptors (e.g., HOG descriptors).

The region matching algorithm of [9], which is an extension of "PatchMatch" [3], is based on the following idea. Each descriptor in each image, *randomly samples* several descriptors in another image and chooses the one with the best match. It then tries to propagate its match (with appropriate shift) to its neighboring descriptors. The neighboring descriptors will change their current match only if the new suggested match is better. Therefore, it is enough for one descriptor in a recurring region to find its correct matching descriptor in another image, and it can then propagate the correct matches to all the other descriptors in that co-occurring region.

Moreover, [9] quantified the number of random samples per descriptor, which are required to guarantee the detec-

[1]Note that our algorithm is not sensitive to the exact vocabulary size. Very few words ($k \sim 100$) suffice to represent well frequent descriptors (smooth patches, vertical/horizontal edges, etc.). Due to the heavy-tail distribution of natural image descriptors, adding more words would only refine the frequent descriptor representatives, and not add the rare ones [6]. Thus, rare descriptors will have a high error w.r.t the codebook regardless of its size.

tion of a co-occurring region between two images with high probability. For example, using 40 random samples per descriptor guarantees the detection of recurring regions of at least 10% of the image size, with very high probability - above 98%. Therefore, large co-occurring regions between two images can be detected at *linear* time.

If shared regions are searched between every pair of images, then the complexity will grow quadratically with the number of images, making it prohibitive for large image collections. Fortunately, [9] showed that *collaboration between images* can resolve this problem, maintaining linear complexity in the size of the collection. This is done by allowing descriptors to randomly sample from the entire image collection, while images "suggest" to each other where to sample and search in the next iteration. This induces a guided random walk with high probability of finding the large shared regions between the images in the collection, at linear time. For more details, see [9].

Incorporating Scale Invariance: In order to handle scale invariance, we generate from each image a cascade of multi-scale images (with scales $1, 0.8, \ldots, 0.8^5$). The region detection algorithm is applied to the entire multi-scale collection of images, allowing shared regions to be detected also between co-objects of different scales.

4. From Co-occurring Regions to Co-segments

The detected non-trivial co-occurring regions induce meaningful affinities between image parts across different images. However, they cover only portions of the co-segments and may cross their boundaries (See Fig 2.b). We use the regions and their affinities to *seed* the co-segments and estimate for each pixel its 'co-segment likelihood'.

4.1. Initializing the Co-segments

Although the detected shared regions do not form 'good' segments on their own, they provide a rough estimation of the location of the co-objects within the image. We use a "soup of segments" (i.e. multiple overlapping segment candidates) to refine and better localize the co-segments. This is done as follows:

1. For each image I, extract a "soup of segments" $\{S_l\}$ using the hierarchal segmentation of [8]. This yields several hundred segments per image, with sizes from small to large. Examples of such segments can be found in Fig 2.c.

2. Compute the *co-segment score* for each segment S_l by its "affinity density", induced by the shared regions:

$$Score(S_l) = \frac{1}{|S_l|} \sum_m \text{Aff}(R_m | I, I_{\chi(m)}) \qquad (5)$$

where $\{R_m\}$ are shared regions detected between image I and other images, with high intersection with segment S_l (at least 75% intersection). $\chi(m)$ is the index of the image in which R_m detected its region match. Summing the contributions of all of these regions and normalizing by the segment size $|S_l|$ results in the "affinity density" of the segment. This allows comparing segments of different sizes.

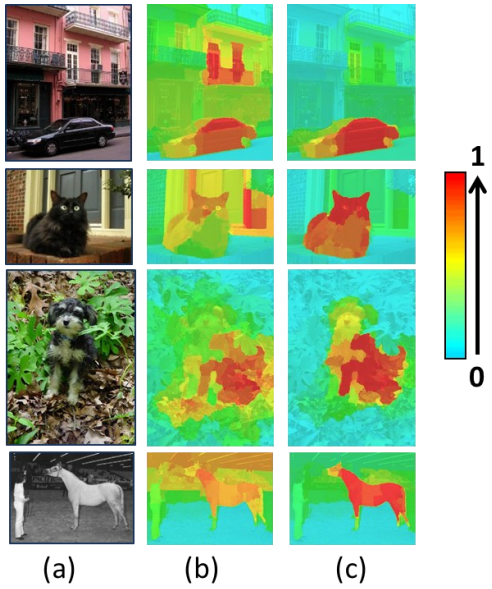

Figure 3. **The co-segmentation likelihood maps**. (a) The input image (b) The initial estimation (Sec. 4.1) (c) The final estimation after 5 iterations of "consensus scoring" (Sec. 4.2).

3. For each pixel p find the K (we use 10) segments $\{S_k\}$ with the highest co-segment scores $\{Score(S_k)\}$ which contain that pixel. Estimate the co-segmentation likelihood per pixel $CSL(p)$ by averaging the co-segment scores of its K best segments:

$$CSL(p) = \frac{1}{K} \sum_k Score(S_k) \qquad (6)$$

4. Normalize the co-segmentation likelihood map of the entire image to be in the range between 0 to 1.

4.2. "Consensus Scoring"

In Sec. 4.1, we have shown how to estimate co-segmentation likelihood maps, induced by detecting statistically significant co-occurring regions for each image in the collection, combined with information about segment boundaries extracted from a "soup of segments". We next show how images can collaborate and share information with each other regrading their co-segmentation likelihood maps to improve the overall quality of the co-segmentation. This is done by allowing images to "score" each other's co-segments. The co-segmentation score is determined using the *consensus* between each region and all its detected co-occurring regions in other images (according to their co-segmentation likelihood maps). We regard this as using the "wisdom of crowds of images" for increasing the fidelity and accuracy of the co-segmentation.

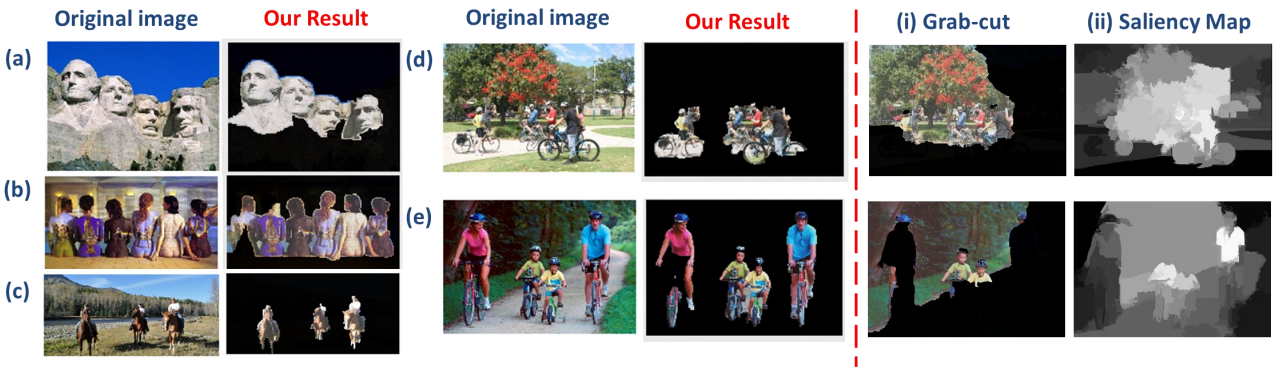

Figure 4. **Co-Segmentations applied to a single image:** Repetitive structures in a single image are detected and co-segmented. The co-segments within each image may have different appearance and scale and may be surrounded by large amounts of clutter. We compare our results to two baselines: (i) Grab-cut initialized with a central window of size 25% of the image (ii) Saliency map of [10]

More precisely, the likelihood of each pixel to be a part of a co-segment increases/decreases according to the *consensus belief* of (i) its spatially surrounding pixels within the same image and (ii) its corresponding pixels in other images, determined by the detected shared regions [2].

Let $p \in I$ be a pixel. Let $p_1, \ldots, p_M \in Neighborhood(p)$ (we use a neighborhood of radius 5 pixels). Let $q_1, \ldots, q_{M'}$ be corresponding pixels to p in all other images induced by the detected shared regions. Then we update the co-segmentation likelihood of each pixel $CSL(p)$ at iteration $(t+1)$ as follows:

$$\log CSL^{(t+1)}(p) = \frac{1}{2} \cdot \frac{1}{M} \cdot \sum_{i=1}^{M} \log CSL^{(t)}(p_i) + \frac{1}{2} \cdot \frac{1}{M'} \cdot \sum_{j=1}^{M'} \log CSL^{(t)}(q_j) \quad (7)$$

We initialize the co-segmentation likelihood of each image (at $t = 0$) using the estimation made in Sec. 4.1 and then perform the re-scoring iteratively (we use 5 iterations). By performing several such scoring phases, we allow regions which are not directly connected to each other to also collaborate and 'share' information regarding the co-segmentation likelihood.

Examples of the estimated co-segmentation likelihood before and after performing the re-scoring iterations can be found in Fig. 3. Note that in the initial co-segmentation likelihood maps there may still remain clutter with high values. Using several iterations of "consensus scoring" suppresses the clutter and reveals the 'true' co-segments. Our experiments show that adding "consensus scoring" yields an improvement of $4\% - 7\%$ in the co-segmentation results.

Obtaining the final co-segments: The above process results in a *continuous* map for each image (with values between 0 and 1). To obtain the final co-segments (i.e., binary co-segmentation maps), we use Grab-cut [15], where the unary terms (background/foreground likelihood) are initialized using our continuous co-segmentation likelihood maps. We use the modified Grab-cut implementation of [14]. Results are shown in Fig. 5 and Tables. 1,2, and are explained and analyzed in Sec. 6

Computational Complexity: Our algorithm is *linear* in the number of images we wish to co-segment, due to the linearity of all its components. This includes linearity of our co-occurring region detection algorithm among *all* images (Sec. 3.2), due to its randomized nature.

5. "Co-segmentation" of a Single Image

To show the power of our approach, we start with an extreme case – co-segmentation of a *single* image. Co-segmentation methods that apply to very few (e.g., 2) images, usually assume high similarity in appearance between the co-objects (e.g., same colors). Handling large variability in appearance between the co-segments usually requires a large number of images, in order to "discover" shared properties of the co-objects (e.g., using unsupervised learning). Thus, a *single* image with few *non-trivial* co-occurrences of an object (such as the examples in Fig. 4), will pose a challenge for existing co-segmentation methods. We next show that our framework can handle even those extreme cases.

When a complex region recurs in the image, and is unlikely to recur at random (i.e., is statistically significant), it provides high evidence that those two image parts should be grouped together (segmented jointly). This is true even if this region never recurs elsewhere again. The co-occurring regions are detected by applying the randomized search and propagation algorithm internally on the image itself. To prevent a trivial composition of a region from itself, we restrict each descriptor to sample descriptors only outside the immediate neighborhood around the descriptor (typically of radius $\frac{1}{16}$ of the image size). The co-segmentation likelihood is estimated the same way as described in Sec. 4 and so is the final extraction of the co-segments. However, here we use "consensus" of co-occurring regions within the same image and not across different images as before. To cope

[2] Recall that when shared regions are detected, each pixel in one region is mapped to a pixel in the other region.

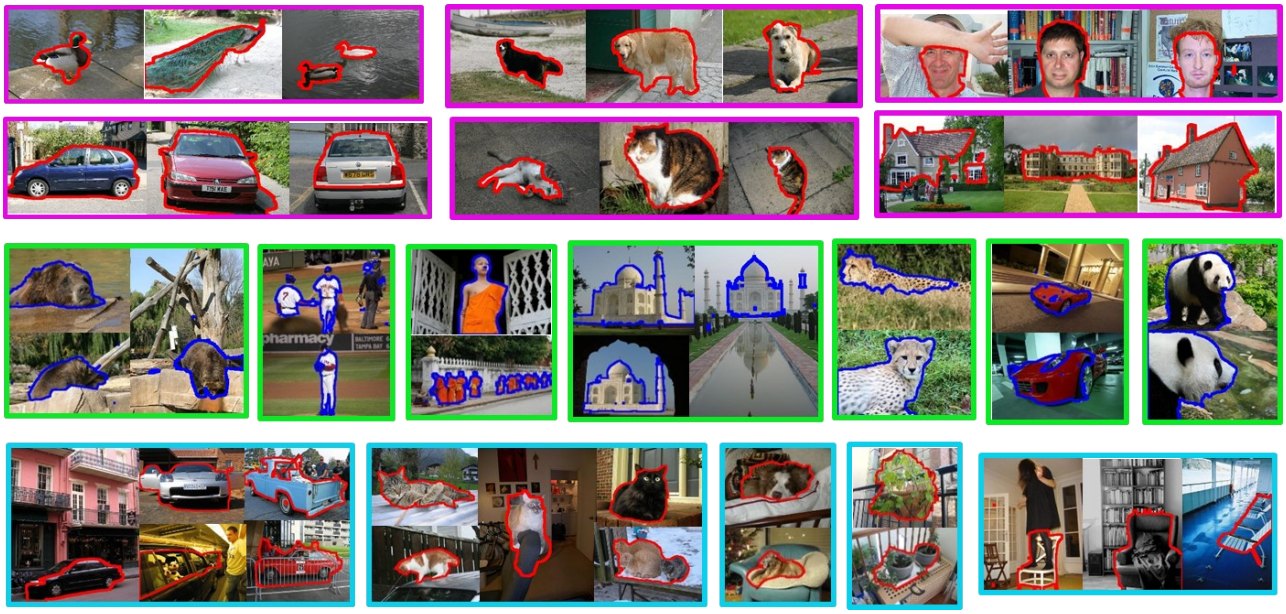

Figure 5. **Co-segmentations produced by our algorithm on MSRC (top), iCoseg (middle) and PASCAL-VOC (bottom)**. In each box, we show co-segmentation results for a few images from a certain class (all the images in the class were used for the co-segmentation).

with scale difference of the co-segments, we search for co-occurring regions across different scales of the same image.

Examples of co-segmentation produced by our algorithm on single images with recurring objects (but with non-trivial recurrences) can be found in Fig. 4. The descriptors used were HOG descriptors. Note that such image segmentation requires no prior knowledge nor any training data beyond the individual segmented image. Moreover, the co-objects segmented in a single image need not repeat in their entirety, and may vary in appearance in their different instances within the image. One co-object can be composed using regions extracted from several other co-segments (possibly at different image scales), thus generating a new configuration.

In a way, this is very similar to the definition of [2], which defines a "good image segment" as one which is easy to compose (like a puzzle) from other regions of the segment, yet is hard to compose from the rest of the image outside the segment. However, unlike [2], we do not require any external user-assisted marking on the desired object to be segmented. Moreover, we do not require that the *entire* co-segment will be composed of other parts of the co-segment, only sufficient portions of it.

Regular image segmentation algorithms will produce more fragmented segmentations of these images. Applying Grab-cut, using an initialization with a central window of size 25% of the image, fails to produce meaningful co-segmentations on such images (see Fig. 4 – second column from the right). Similarly, saliency based segmentation will not suffice either, since the co-occuring object is not necessarily salient in the image, and there can be other salient image parts (e.g. the red tree in Fig. 4d – the saliency maps were generated using [10]).

We, on the other hand, are able to produce good co-segmentations of these images by employing the reoccurrence of large non-trivial regions within each image. As can be seen in Fig. 4, we are able to co-segment the president faces in Mount Rushmore (although their color is similar to the the rest of the mountain) and the women on the Pink-Floyd wallpaper (despite their different appearance). Moreover, our co-segments can be disconnected segments (such as the bikes and horse riders), and may have difference in scale (such as the child and his parents in Fig. 4e).

6. Experimental Results

We empirically tested our algorithm on various datasets, including evaluation datasets (MSRC, iCoseg), on which we compared results to others, as well as more difficult datasets (PASCAL), on which to-date *no* results were reported for *unsupervised* co-segmentation.

6.1. Experiments on Benchmark Datasets

We used existing benchmark evaluation datasets (MSRC, iCoseg) to compare results against [11, 21, 13, 14, 18, 17] using their experimental setting and measures - see Table 1. The co-segmentation is performed on each class of each dataset separately and we report the average performance of all classes in each dataset. There are two common measures for co-segmentation performance: (a) *"average Precision"* **(P)** - the percentage of correctly segmented pixels (of both the foreground and background pixels) (b) *"Jaccard index"* **(J)** - the intersection divided by the union of the

iCoseg	Ours	[14]	[17]	Joint Grab-Cut	Grab-Cut [15]
P	92.8%	91.4%	89.8%	88.2%	82.4%
J	0.73	-	0.69	-	-

iCoseg subset	Ours	[17]	[21]	[18]	[11]
P	94.4%	89.6%	85.4%	83.9%	78.9%
J	0.79	0.68	0.62	-	-

MSRC	Ours	[17]	[11]	[13]
P	89.2%	87.7%	73.6%	54.6%
J	0.73	0.68	0.5	0.37

MSRC subset	Ours	[17]	[21]
P	92%	92.2%	90.2%
J	0.77	0.75	0.71

Table 1. **Comparison to previous co-segmentation methods on benchmark datasets.** P and J denote the average Precision and average Jaccard index, respectively.

PASCAL		Ours	Grab-Cut [15]	best proposal of [8]	[11]
All	P	84%	76%	60.4%	59.5%
Subset	P	86.8%	78.9%	63.5%	60.8%
All	J	0.46	0.38	0.3	0.23
Subset	J	0.53	0.45	0.35	0.25

Table 2. **Performance evaluation of co-segmentation on PASCAL-VOC 2010**. P and J denote the average Precision and average Jaccard index, respectively.

co-segmentation result with the ground truth segmentation. The Jaccard index reflects more reliably the true quality of the co-segmentation. We report both measures wherever the information was available.

The MSRC dataset [20] contains 420 images from 14 classes. The co-segments in each class have color, pose and scale differences, yet tend to be quite salient in the image with few clutter. The iCoseg dataset [4] consists of 643 images from 38 classes. The co-segments in each class typically have similar color properties, but may occur at different poses and scales. Visual examples of co-segmentation results on MSRC and iCoseg can be found in Fig. 5.

For the MSRC dataset, we used dense HOG (Histogram of Oriented Gradients) descriptors in order to be invariant to the large difference in appearance of each class. For the iCoseg dataset we added color descriptors in addition to the HOG descriptors (we used densely sampled descriptors, which are concatenation of HOG and LAB color histograms). The color descriptors were added here to leverage on the color similarity of the co-segments within each class, which is a characteristic of the iCoseg dataset. We built a descriptor dictionary for each class separately and used it to compute the error of each descriptor with respect to the dictionary, which is required in the affinity calculation (Eq(4)).

We obtain state-of-the-art results on the MSRC dataset, obtaining 50% and 100% improvement in Jaccard index over [11] and [13], and 7% improvement over [17]. We also compared to [17, 21] on the subset of MSRC that they used (7 classes with 10 images per class). Our results in Jaccard index are slightly better than [17], and 9% better than [21].

In the iCoseg dataset, in order to obtain the final *binary* co-segments from our continuous likelihood maps, we followed the 'Joint-Grab-Cut' suggestion of [14]. Namely, instead of applying Grab-cut to each image separately, it is applied jointly to all the images (initializing and updating the color models to all images at once). This makes sense since the co-segments in each class of iCoseg are known to have similar color properties. We initialize the 'Joint-Grab-Cut' with our continuous co-segmentation likelihood maps.

We obtain state-of-the-art results also on the iCoseg dataset, exceeding [17, 21, 18, 11] on the subset they used (16 classes), by a significant gap. We obtain an improvement of 16% and 27% in Jaccard index over the previous state-of-the-arts [17] and [21]. Our results on the full iCoseg dataset are much better than applying a baseline of Grab-cut or Joint-Grab-Cut, when these are initialized with a central window of size 25% of the image. Moreover, our results on the full dataset are better than [17] and even slightly better than [14], even though [14] relied on knowledge transfer from an external dataset with human-annotated segmentations of objects, whereas we do not. This shows that there exists enough internal information for co-segmentation within the collection of images alone.

6.2. Experiments on the PASCAL-VOC Dataset

The PASCAL dataset is a very challenging dataset for object co-segmentation, due to the large variability in object scale, appearance, and due to the large amount of distracting background clutter. Since the co-objects are so different, simple models of co-objects will not suffice here. Moreover, initializing the co-segmentation using saliency maps will also be problematic, since the co-segments are not necessarily salient in the image, as there are many other distracting objects in the image. To the best of our knowledge, to-date, *no* results were reported on PASCAL for *purely unsupervised* co-segmentation.

We made a first such attempt, restricting ourselves to images from PASCAL-VOC 2010, in which at least one of the co-objects is not labeled as 'difficult' or 'truncated', and the total size of the co-object is at least 1% of the image size. We remain with 1037 images from the 20 PASCAL classes. We split the classes into two subsets - the first consists of animal and vehicle classes (total of 13 classes) and the second consists of the remaining classes such as person, table and potted plant (total of 7 classes). The second subset seems much harder (as indeed verified in our experiments) since there is a lot of co-occurring clutter and objects in those classes in addition to the main co-occurring object. This makes the co-segmentation problem much more ill-posed.

For the PASCAL dataset, we used dense HOG descriptors in order to be invariant to the large difference in appearance of each class. On the first PASCAL subset, we obtain a mean segmentation Precision of 86.8%. This seems quite good compared to results reported on the easier MSRC and iCoseg datsets. However, observing the Jaccard index of our co-segmentation, we can see that there is still place for improvement: our algorithm obtains performance of 0.53, whereas on MSRC and iCoseg we obtained performance of 0.73. When adding the remaining 7 classes, the segmentation Precision drops to 84% and Jaccard index to 0.46. The reason for this large gap between the Precision and Jaccard index measures is that Precision gives equal contribution to foreground and background, whereas the Jaccard index considers only the foreground. In PASCAL, the background is $> 90\%$ of the image size and the foreground is $< 10\%$.

Examples of some successful co-segmentation results on PASCAL can be found in Fig 5. Notice the large amount of clutter and distracting objects which exist in those images - yet our algorithm yields very good results. Moreover, even for difficult classes like Potted-Plant or Chair, our algorithm is sometimes able to produce very appealing results.

We further generated three baseline comparisons on this PASCAL dataset - see Table 2. The first was generated using the co-segmentation method of [11]. The two other were methods which segment each image separately: (a) Grab-Cut initialized with a central window of size 25% of the image. (b) [8]'s best object proposal. The best results among all *baselines* were obtained, perhaps surprisingly, by Grab-Cut - mean segmentation Precision of 76% (Jaccard index of 0.38). This can be explained by the fact that even in a challenging dataset like PASCAL, the main object is still located at the image center at quite many images. Our results were superior to all the baseline methods.

Failure Cases: In our failure cases, the co-object is usually not missed, but distracting background is added to it. This typically occurs when the background contains objects which recur in multiple images (other than the co-object). For example, in the PASCAL 'Chair' class there are lots of tables, so these are also co-segmented along with the chairs. Examples of failure cases can be found in our **project website** `www.wisdom.weizmann.ac.il/~vision/CoSegmentationByComposition.html`.

7. Conclusion

In this paper we suggest a new approach to object co-segmentation. We define 'good' co-segments to be ones which can be easily composed from large pieces of other co-segments, yet are difficult to compose from the remaining image parts. This enables co-segmentation of very challenging scenarios, including co-segmentation of *individual* images containing multiple *non-trivial* reoccurrences of an object, as well as challenging datasets like PASCAL.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. Classcut for unsupervised class segmentation. In *ECCV*, 2010.

[2] S. Bagon, O. Boiman, and M. Irani. What is a good image segment? a unified approach to segment extraction. In *ECCV*, 2008.

[3] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. In *SIGGRAPH*, 2009.

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

[5] O. Boiman and M. Irani. Similarity by composition. In *NIPS*, 2006.

[6] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *CVPR*, 2008.

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

[8] I. Endres and D. Hoiem. Category independent object proposals. In *ECCV*, 2010.

[9] A. Faktor and M. Irani. "clustering by composition" - unsupervised discovery of image categories. In *ECCV*, 2012.

[10] H. Jiang, J. Wang, Z. Yuan, T. Liu, N. Zheng, and S. Li. Automatic salient object segmentation based on context and shape prior. In *BMVC*, 2012.

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

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

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

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

[15] C. Rother, V. Kolmogorov, and A. Blake. Grabcut - interactive foreground extraction using iterated graph cuts. In *SIGRAPH*, 2004.

[16] C. Rother, V. Kolmogorov, T. Minka, and A. Blake. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. In *CVPR*, 2006.

[17] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu. Unsupervised joint object discovery and segmentation in internet images. In *CVPR*, 2013.

[18] J. Rubio, J. Serrat, and A. Lopez. Unsupervised co-segmentation through region matching. In *CVPR*, 2012.

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

[20] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, 2006.

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

[22] J. Winn and N. Jojic. Locus: learning object classes with unsupervised segmentation. In *ICCV*, 2005.