Object-based RGBD Image Co-segmentation with Mutex Constraint

Huazhu Fu[1] Dong Xu[1] Stephen Lin[2] Jiang Liu[3]
[1]School of Computer Engineering, Nanyang Technological University, Singapore
[2]Microsoft Research, Beijing, China
[3]Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore

Abstract

We present an object-based co-segmentation method that takes advantage of depth data and is able to correctly handle noisy images in which the common foreground object is missing. With RGBD images, our method utilizes the depth channel to enhance identification of similar foreground objects via a proposed RGBD co-saliency map, as well as to improve detection of object-like regions and provide depth-based local features for region comparison. To accurately deal with noisy images where the common object appears more than or less than once, we formulate co-segmentation in a fully-connected graph structure together with mutual exclusion (mutex) constraints that prevent improper solutions. Experiments show that this object-based RGBD co-segmentation with mutex constraints outperforms related techniques on an RGBD co-segmentation dataset, while effectively processing noisy images. Moreover, we show that this method also provides performance comparable to state-of-the-art RGB co-segmentation techniques on regular RGB images with depth maps estimated from them.

1. Introduction

The goal of co-segmentation is to extract similar foreground objects from among a set of related images [30, 17, 19, 36, 31]. In contrast to single-image segmentation, co-segmentation makes use of the information in multiple images to infer the objects to extract. Existing methods operate on RGB images and utilize descriptors such as color histograms, SIFT and HOG to perform co-segmentation. However, color-based features have limitations, as they cannot distinguish a foreground from a similarly colored background, and are sensitive to illumination differences among images. These issues are illustrated in Fig. 1(c), where the common foreground object is merged with a background object of similar color in the second row, and illumination change causes the target to be missed in the third row.

To address this problem, we propose in this paper to introduce the depth cue into co-segmentation, which can help to reduce ambiguities with color descriptors. How to effectively utilize depth information in co-segmentation is not straightforward. In single RGBD image segmentation, depth can be treated as an additional color channel, since the depth over a foreground object is generally consistent yet distinct from the background [25]. However, in co-segmentation where commonalities among images are exploited, different depth values for the same object in different images can create matching problems.

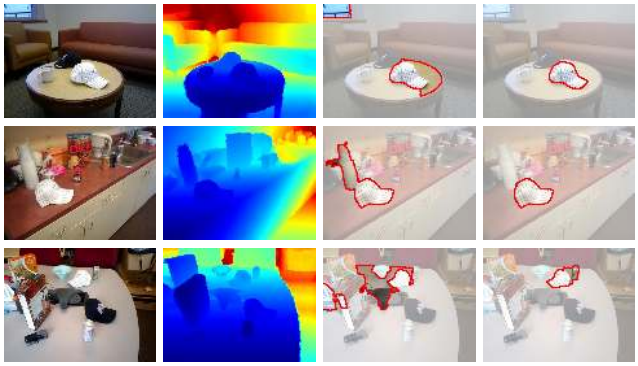

Figure 1. Co-segmentation with RGB vs. RGBD images. (a) A set of RGB images which all contain a white cap in common. (b) The corresponding depth maps. (c) Co-segmentation results based on RGB images, which exhibit errors due to similarly colored background objects (second row) or illumination change (third row). (d) Co-segmentation results with RGBD images. Our use of depth cues notably improves co-segmentation quality.

In this paper, we present an object-based RGBD image co-segmentation method based on *RGBD co-saliency maps*, which capitalize on depth cues to enhance identification of common foreground objects among images. Depth is also utilized to provide additional local features for region comparison and to improve selection of object-like regions [14]. Objectness has been used in co-segmentation to overcome limitations of low-level features in separating complex foregrounds and backgrounds [35], but such methods have been formulated with an assumption that *exactly one* common object exists in *all* of the images. If the common foreground

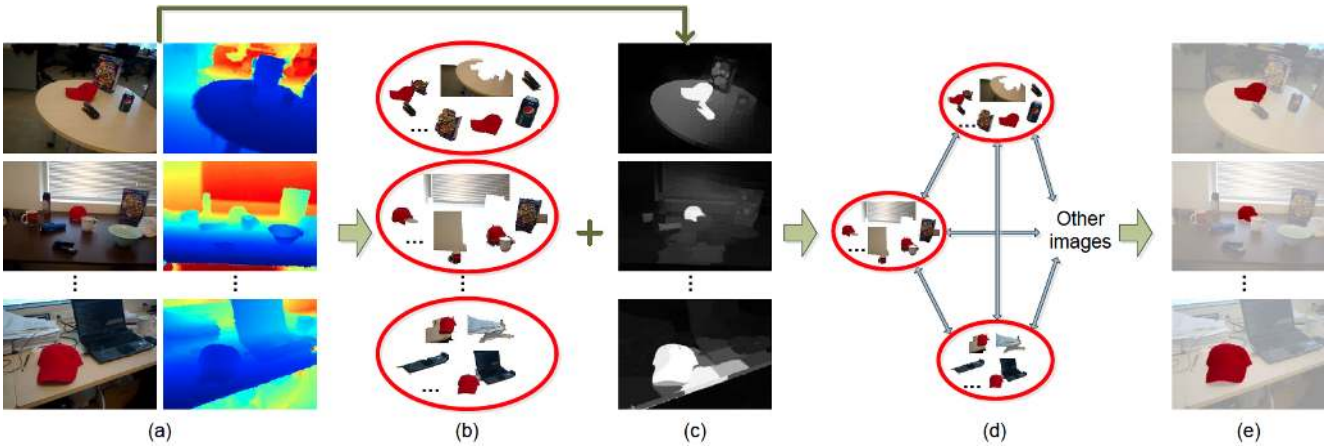

Figure 2. Overview of our approach. From the input RGB images and their depth maps (a), we generate the object candidates (b) and RGBD co-saliency maps (c). An co-segmentation graph (d) is built to select the best candidates as the co-segmentation result (e).

is missing in an image, an irrelevant region will be extracted instead. In our work, we additionally address this issue through a fully-connected graph formulation that enables the option of selecting no regions or more than one region in an image with the help of the opposite effects of the unary and pairwise terms.

An overview of our approach is shown in Fig. 2. Given a set of images and their depth maps, we first generate a foreground candidate pool for each image using RGBD-based multiscale combinatorial grouping [14]. For each candidate region, an RGBD co-saliency score is computed and added to the RGBD objectness score calculated in candidate generation to measure the likelihood that the candidate belongs to the foreground. With the candidates and their likelihood scores, a candidate selection graph is built, with each node representing a candidate in an image, and pairwise edges added to connect all pairs of candidates among all of the images. Mutual exclusion (mutex) constraints are also introduced between nodes to restrict candidate selection within the same image. The graph is formulated as a binary integer quadratic program (IQP) problem, which is optimized by using the fixed-point iteration technique.

To the best of our knowledge, this is the first paper to address co-segmentation in RGBD images. We demonstrate that depth cues can serve as a helpful complement to color features in co-segmentation. Moreover, our formulation with mutex constraints is shown to provide greater flexibility by handling any number of common foreground objects in an image. For evaluation, we constructed a new RGBD co-segmentation dataset with pixel-level ground-truth. In addition to co-segmentation of RGBD images, we also show that our technique can improve co-segmentation results for regular RGB images by incorporating depth estimated from the images using [18].

2. Related work

RGBD-based segmentation has attracted much interest because of the wide availability of affordable RGBD sensors. Supervised RGBD segmentation methods employ depth cues as region descriptors for object classes, and learn object class models from labeled training data [33, 14]. In unsupervised RGBD segmentation, depth cues are used to better preserve object boundaries and to constrain the object surface to be smooth and consistent [25, 29]. While these uses of depth are suitable for precise region extraction in single-image segmentation, the co-segmentation task addressed in our work is instead driven by relationships among regions in different images. For co-segmentation, we utilize depth information in a manner that helps to infer the common foreground in a set of images, in part through the formulation of RGBD co-saliency maps. From the examination of similar object regions in different images, accurate segmentation results are obtained without the need for labeled training data.

Co-segmentation was first introduced in [30], which uses histogram matching to simultaneously segment the common object in a pair of images. This paper has inspired much follow-up work, including co-segmentation methods based on maximum flow optimization to increase efficiency [15], intelligent scribble guidance to facilitate processing of larger image sets [2], discriminative clustering that handles greater variation in foreground appearance [17], and rank constraints for scale invariance [26]. In [35], an object-based framework was introduced for co-segmentation, in which a measure of 'objectness' is considered in identifying foregrounds. Objectness was also utilized for co-segmentation in [24] within a shortest path search framework. Different from methods based on low-level descriptors, object-based techniques make use of a mid-level representation that aims to delineate an object's entirety. Our

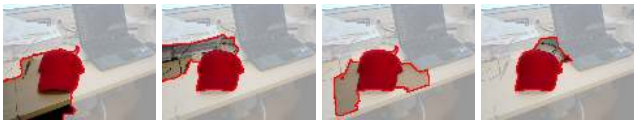

Figure 3. Some samples of the object candidates obtained with [14]. It can be seen that candidates within an image may have substantial overlap.

method is also based on objectness, but in contrast to previous object-based co-segmentation techniques, it utilizes depth information to improve the detection of object regions and to enhance identification of common foreground objects in the images.

Object-based co-segmentation techniques typically employ object proposal methods [7, 34, 11, 16, 20] to generate a pool of foreground candidates for each image. Among these candidates, the co-segmentation result is determined primarily by its commonality with candidates in other images. This is formulated in [35, 24] as a graph where a layer of nodes represents the candidates in an image, and links that represent pairwise commonality energy are placed between each pair of nodes in different layers. With this graph structure, a co-segmentation solution will include exactly one candidate per image. The work in [13] extracts common candidates from multiple videos by using an object-based co-selection graph, which is formulated as a classical CRF energy minimization problem. However, these methods have an important assumption that the common foreground objects must appear in all the images/videos in the set. This strong assumption greatly limits the application field of these methods. If the common foreground object happens to be missing in one of the images, however, these methods will segment an irrelevant region instead. In our work, we propose a more general graph structure where links also exist among nodes of the same image, and an arbitrary number of candidates within an image can potentially be chosen, including no candidates at all. With such a graph, it is possible to obtain inapt solutions such as multiple overlapping candidates within the same image as shown in Fig. 3, which would have significant pairwise commonality. To avoid such solutions, we make use of mutex constraints among candidates within the same image.

Mutex constraints express mutual exclusion rules where if one candidate is selected, certain other candidates cannot also be chosen. These have been used in object-based video segmentation to prevent selection of spatially distant objects in consecutive frames, and more than one object in a single frame [23]. In our work, we also utilize mutex constraints, but instead they are used to avoid selection of overlapping candidates in the co-segmentation result. With the use of mutex constraints together with our fully-connected graph structure, it is possible for our co-segmentation to select no candidates in an image when the common foreground object is missing. Moreover, it is also possible to select multiple non-overlapping candidates if there exist multiple similar instances of the common foreground object in an image.

3. Proposed method

Given a set of RGBD images $\{I^1, ..., I^N\}$, we first generate a set of object candidates, denoted by $\{x_1, ..., x_M\}$. In our work, the candidates are computed using the 2.5D region proposals generation method in [14], which is based on multiscale combinatorial grouping [1] with the use of depth cues. Our goal is to discover a small subset of candidates that contain the same or similar foreground object among the images. Thus we introduce a binary label variable u_i for each object candidate x_i, which takes either the foreground label $u_i = 1$, or the background label $u_i = 0$. We formulate the task of object-based co-segmentation as a labeling problem in a weighted graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$, in which $\mathcal{V} = \{x_1, ..., x_M\}$ is the set of $\|\mathcal{V}\| = M$ nodes representing the object candidates in all of the images. The edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ are represented in a symmetric pairwise matrix with all nonnegative entries. Moreover, a mutex constraint modeled as a binary matrix $\mathbf{M} \in \{0,1\}^{M \times M}$ is added to connect the nodes in the graph. If $M(i,j) = 1$ then the two nodes u_i, u_j cannot belong to the same label. For all vertices u_i, we set $M(i,i) = 0$. The goal of our RGBD co-segmentation is to find a labeling $\mathbf{u} = [u_1, ..., u_M]^T$ that minimizes the following objective function:

$$\begin{aligned} \mathbf{u}^* &= \arg\min_{\mathbf{u}} \frac{1}{2}\alpha \mathbf{u}^T \mathbf{A}\mathbf{u} - \mathbf{b}^T\mathbf{u}, \\ &\text{s.t. } \mathbf{u}^T\mathbf{M}\mathbf{u} = 0, \text{ and } \forall i \in \mathcal{V} : u_i \in \{0,1\}, \end{aligned} \quad (1)$$

where $\mathbf{A}$ is the pairwise matrix, and $\mathbf{b}$ associates a positive unary term b_i to each node u_i.

3.1. Unary term

The unary term $\mathbf{b}$ measures the likelihood that the candidate belongs to the foreground, and it is defined as

$$b_i = Obj(u_i) \cdot Sal(u_i). \quad (2)$$

The first term is a 2.5D objectness score $Obj(u_i)$ computed from [14], which reflects the confidence that a region contains a generic object in the RGBD image. The second term is the RGBD co-saliency score $Sal(u_i)$.

Co-saliency detection relates to visually salient stimuli combined with consistency among multiple images [8, 12, 6, 5]. It has been shown to be helpful for discovering a common foreground in an image set. However, existing co-saliency detection methods [8, 12, 6] are based only on color images, and can easily be misled by complex backgrounds as shown in Fig. 4. On the other hand, depth-based saliency methods [27, 22, 28] can effectively distinguish salient objects from backgrounds of similar color,

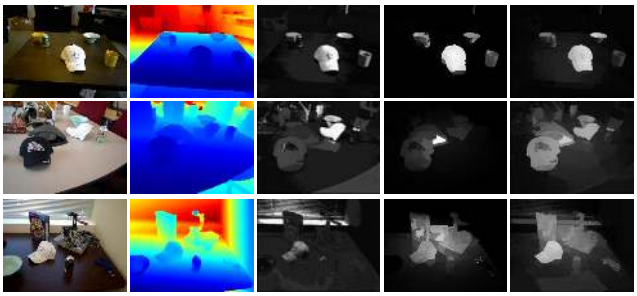

Figure 4. Saliency map comparisons for a set of images (a) and their corresponding depth maps (b). In (c), color-based co-saliency detection (e.g., [12]) mixes the common foreground (white cap) with the complex backgrounds. In (d), RGBD saliency detection (e.g., [28]) generally distinguishes the foreground from the complex background, but it lacks a way to identify common objects in multiple images. By contrast, our RGBD co-saliency map in (e) effectively combines depth and co-saliency cues to detect the common foregrounds in multiple RGBD images.

but do not account for commonalities among images. In our work, we take advantage of both approaches to combine the depth and co-saliency cues from multiple RGBD images by integrating them within the saliency map fusion framework of [6]. The method in [6] exploits the low-rank relationship of multiple saliency sub-maps and obtains self-adaptive weights to generate a final co-saliency map. We combine five kinds of sub-maps, specifically single-image RGB saliency [37, 38] and RGB co-saliency [12] and RGB-D saliency [28, 9]. This yields the RGBD co-saliency map, as shown in Fig. 4, which measures the likelihood of belonging to the foreground while accounting for object similarity among the images. From the RGBD co-saliency maps, we compute the RGBD co-saliency score $Sal(u_i)$ for node u_i as

$$Sal(u_i) = S(u_i) + \log\left(\frac{S(u_i)}{S(\bar{u}_i)} + 1\right), \tag{3}$$

where $S(u_i)$ denotes the mean RGBD co-saliency map value among the pixels of object candidate u_i, and $\bar{u}_i$ denotes the pixels outside of candidate u_i but within the minimum bounding box enclosing the candidate. Our RGB-D co-saliency score $Sal(u_i)$ accounts for two factors, one being the RGBD co-saliency values of the candidate itself, and the other being the regional contrast of the candidate from its surroundings. Different from the objectness score $Obj(u_i)$, which is designed to identify extracted regions that are object-like and compact, the RGBD co-saliency score $Sal(u_i)$ is used to find regions of interest that exist in common among the images.

3.2. Pairwise matrix

The pairwise matrix $\mathbf{A}$ measures the similarity between two object candidates, and is defined as

$$A(i,j) = D_{color}(i,j) \cdot D_{shape}(i,j) \cdot D_{depth}(i,j), \tag{4}$$

where $D_{color}(i,j)$, $D_{shape}(i,j)$, and $D_{depth}(i,j)$ denote the distances between the candidates u_i and u_j based on color, shape and depth features. The RGB histogram and HOG [10] are employed as the color and shape features. The depth kernel descriptor [3] based on depth gradients is used as the depth feature. The feature distances are measured using the L2-norm distance.

3.3. Mutex constraints

The mutex constraint $\mathbf{M}$ is used to control the selection of overlapping candidates within the same image. It is a necessary component for our graph structure in which all candidates from all images are connected to each other. Since significantly overlapping candidates within an image often have a low pairwise distance, they may be selected together in the co-segmentation result if a mutex constraint is not applied between them.

In this paper, we measure the overlap between two candidates as

$$Overlap(i,j) = \frac{R(u_i) \cap R(u_j)}{\min\left(R(u_i), R(u_j)\right)}, \tag{5}$$

where $R(u_i)$ denotes the area of candidate u_i. Based on this overlap measure, we define the mutex constraint matrix as

$$M(i,j) = \begin{cases} 1, & \text{if } Overlap(i,j) \geq \tau \text{ and } i \neq j \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

where τ is an overlap threshold (we set $\tau = 0.2$ in our experiments).

With this mutex constraint and our fully-connected graph structure, it is possible to select any number of candidates, including none, from an image as long as they do not overlap substantially. By setting the overlap threshold to $\tau = 0$, the solution would instead be constrained to select up to one candidate per image, since mutex constraints would exist between each pair of candidates in the image. Our use of mutex is different from that in [23], which constrains the result to have exactly one candidate in each image, and thus cannot deal with cases of missing or multiple common foregrounds.

We note that even though any number of candidates can be selected from an image in our method, trivial solutions are avoided because of the opposite effects of the unary and pairwise terms in Eq. (1). Solutions that select all the candidates are avoided because of the pairwise term, which penalizes differences among the chosen candidates. The mutex constraint helps as well by excluding candidates that are

highly overlapping. On the other hand, solutions that do not select any candidates are prevented by the unary term, which favors selecting as many candidates as possible. Despite this unary term, no candidate will be selected in an image that is missing the common foreground, otherwise the improperly chosen candidate would have high pairwise costs for all its links to the common object in the other images.

3.4. Inference

To infer the co-segmentation solution, we formulate Eq. (1) as an integer quadratic program (IQP). We first combine the mutex constraints into the pairwise matrix to obtain the following objective function:

$$\mathbf{u}^* = \arg\min_{\mathbf{u}} \frac{1}{2}\mathbf{u}^T\mathbf{W}\mathbf{u} - \mathbf{b}^T\mathbf{u}, \quad \text{s.t. } \forall i \in \mathcal{V} : u_i \in \{0,1\}, \tag{7}$$

with

$$\mathbf{W} = \alpha\mathbf{A} + \gamma\mathbf{M}. \tag{8}$$

Similar to [4], we solve the IQP problem by employing the fixed-point iteration method. Let us denote the objective function of Eq. (7) as $f(\mathbf{u}) = \frac{1}{2}\mathbf{u}^T\mathbf{W}\mathbf{u} - \mathbf{b}^T\mathbf{u}$, and let $\mathbf{y} \in [0,1]^M$ denote a discrete point in the continuous domain. We visit a sequence of points $\{\mathbf{y}^{(t)} \in [0,1]^M\}_{t=1,2,\ldots}$ to find candidate solutions for $\mathbf{u}^*$. Each iteration consists of two steps. First, for each point $\mathbf{y} \in [0,1]^M$ in the neighborhood of $\mathbf{y}^{(t)}$, we compute the first-order Taylor approximation of $f(\mathbf{y})$ as

$$\begin{aligned} f(\mathbf{y}) &\approx f(\mathbf{y}^{(t)}) + (\mathbf{y} - \mathbf{y}^{(t)})^T(\mathbf{W}\mathbf{y}^{(t)} - \mathbf{b}) \\ &= \mathbf{y}^T(\mathbf{W}\mathbf{y}^{(t)} - \mathbf{b}) + const, \end{aligned} \tag{9}$$

where $const$ does not depend on $\mathbf{y}$. Since the approximation in Eq. (9) is convex in $\mathbf{y}$, it can easily be computed with a discrete minimizer as

$$\tilde{\mathbf{u}} = \arg\min_{\mathbf{y}} \mathbf{y}^T(\mathbf{W}\mathbf{y}^{(t)} - \mathbf{b}), \tag{10}$$

and

$$\tilde{u}_i = \begin{cases} 1, & \text{if } (\mathbf{W}\mathbf{y}^{(t)} - \mathbf{b})_i \le 0 \\ 0, & \text{Otherwise.} \end{cases} \tag{11}$$

In the second step of iteration t, the algorithm checks whether $\tilde{\mathbf{u}}$ can be accepted as a valid discrete solution if the objective value f decreases. If $f(\tilde{\mathbf{u}}) < f(\mathbf{y}^{(t)})$, we let $\mathbf{y}^{(t+1)} = \tilde{\mathbf{u}}$. In the case that $f(\tilde{\mathbf{u}}) \ge f(\mathbf{y}^{(t)})$, there is a local minimum of f in the neighborhood of points $\mathbf{y}^{(t)}$ and $\tilde{\mathbf{u}}$. We then estimate the local minimizer of f in the continuous domain by linear interpolation:

$$\mathbf{y}^{(t+1)} = \mathbf{y}^{(t)} + \eta(\tilde{\mathbf{u}} - \mathbf{y}^{(t)}), \tag{12}$$

where the optimal value of $\eta \in [0,1]$ is computed by

$$\eta = \begin{cases} 1, & \text{if } \eta^* > 1 \\ \eta^*, & \text{if } 1 \ge \eta^* \ge 0 \\ 0, & \text{if } \eta^* < 0 \end{cases} \tag{13}$$

with

$$\eta^* = \frac{(\mathbf{W}\mathbf{y}^{(t)} - \mathbf{b})^T(\tilde{\mathbf{x}} - \mathbf{y}^{(t)})}{(\tilde{\mathbf{u}} - \mathbf{y}^{(t)})^T\mathbf{W}(\tilde{\mathbf{u}} - \mathbf{y}^{(t)})}. \tag{14}$$

The iterations stop when the following condition is satisfied for all nodes $\tilde{u}_i$:

$$\begin{aligned} &\text{if } (\mathbf{W}\tilde{\mathbf{u}} - \mathbf{b})_i \le 0, \text{then } \tilde{u}_i = 1, \\ &\text{if } (\mathbf{W}\tilde{\mathbf{u}} - \mathbf{b})_i > 0, \text{then } \tilde{u}_i = 0. \end{aligned}$$

After convergence, the last discrete solution $\tilde{\mathbf{u}}$ is used as the final solution of Eq. (7). We initialize $\mathbf{y}^{(0)} \in \mathbb{R}^M$ randomly, where $y_i^{(0)} \in \{0,1\}$ and $\mathbf{y}^{(0)} \neq \mathbf{0}$. Because in our algorithm, the objective value f decreases in each iteration, it converges to a minimum. With a sufficiently large γ in Eq. (7), the obtained solution satisfies the mutex constraints.

On our workstation with an Intel Xeon 3.2GHz CPU and 16GB RAM, it takes about 30s to process a 10k node graph using unoptimized C++ code (for a set of 100 images with 100 candidates/image).

4. Experiments

In our implementation, we generate the top 100 object candidates for each image. We set $\alpha = 0.25$ and $\gamma = 100$ in Eq. (8) as the default parameters for all the experiments.

4.1. RGBD co-segmentation dataset

Since no datasets for RGBD co-segmentation are publicly available, we have constructed one ourselves with some images from the RGBD Scenes Dataset [21] and some that were captured by ourselves. The dataset contains 16 image sets, each of 6 to 17 images taken from indoor scenes with one common foreground object (193 images in total). Pixel-level ground-truth is labeled for the common foreground object in each image[1].

We compare our algorithm with three recent methods for RGB co-segmentation [17, 19, 24]. For all methods, we used the original implementations provided by the authors. Since the method in [24] is also an object-based technique, we additionally include a variant of it (referred to as [24] + Depth) incorporating depth cues in the same way as in our method (with 2.5D proposals, our RGBD co-saliency map, and the depth kernel descriptor [3] included in the pairwise term). For our algorithm, we show not only its final full results (*Our RGBD*), but also those of a few variants: without using depth (*Our RGB*), without the mutex

[1] https://sites.google.com/site/huazhufu/home/rgbdseg

Set (# images)	[17]	[19]	[24]	[24] + Depth	Our RGB w/o M	Our RGB	Our RGBD w/o M	Our RGBD
Ball (13)	66.0 (28.5)	92.6 (57.8)	89.5 (61.0)	**95.2** (**70.2**)	90.7 (34.0)	90.5 (55.9)	86.2 (28.6)	93.0 (48.2)
Blue light (7)	45.7 (1.4)	89.1 (7.1)	91.9 (28.7)	91.9 (28.7)	90.1 (26.9)	**92.0** (**33.2**)	91.3 (30.1)	89.1 (31.5)
Box (20)	65.0 (22.6)	92.0 (65.6)	91.2 (38.0)	**92.4** (49.2)	90.8 (32.8)	85.4 (30.6)	91.0 (**66.5**)	89.4 (34.8)
Carving (14)	50.6 (19.5)	85.8 (40.3)	86.2 (35.0)	86.6 (36.1)	90.6 (29.8)	82.4 (34.3)	87.8 (22.4)	**91.6** (**45.6**)
Computer (8)	73.0 (47.1)	**90.0** (**65.9**)	82.5 (59.9)	84.6 (53.7)	80.5 (25.9)	82.2 (40.7)	86.7 (53.2)	87.4 (59.9)
Cyan bowl (18)	49.5 (2.8)	90.0 (26.4)	80.9 (17.1)	88.9 (24.3)	93.3 (16.3)	93.6 (**32.2**)	91.4 (10.8)	**95.3** (31.6)
Green bowl (15)	53.5 (3.0)	88.6 (16.1)	85.2 (22.1)	**90.7** (**38.3**)	87.2 (20.1)	90.4 (29.2)	89.4 (21.5)	86.8 (15.7)
Person (8)	83.4 (63.7)	85.6 (66.9)	72.8 (47.8)	75.3 (33.2)	87.3 (60.7)	83.5 (61.8)	89.4 (66.8)	**91.6** (**78.4**)
Red cap (13)	51.8 (10.0)	91.1 (39.2)	86.1 (52.7)	93.7 (62.5)	95.6 (22.3)	97.0 (**74.3**)	92.3 (20.5)	**97.6** (59.5)
Red light (9)	48.3 (1.9)	67.8 (2.1)	87.1 (9.2)	88.8 (15.8)	**94.8** (30.8)	93.7 (22.6)	89.7 (41.7)	94.1 (**43.4**)
Shoes (6)	61.3 (21.5)	96.1 (68.0)	97.3 (83.5)	**98.3** (**86.5**)	94.3 (86.5)	93.5 (46.5)	97.6 (85.5)	**98.3** (**86.5**)
Soda can (7)	50.8 (2.5)	95.2 (8.5)	78.5 (29.4)	85.3 (19.4)	**95.6** (66.3)	89.9 (20.1)	97.0 (56.8)	99.3 (**77.6**)
Vase (17)	49.0 (5.5)	52.6 (2.6)	78.2 (12.6)	86.1 (20.1)	96.0 (24.9)	92.4 (30.6)	93.8 (29.5)	**96.9** (**53.6**)
White bowl (12)	45.8 (0.03)	88.0 (19.6)	78.3 (5.4)	80.6 (4.9)	86.6 (15.1)	85.7 (17.4)	92.0 (11.5)	**92.7** (**26.9**)
White cap (15)	47.3 (1.0)	76.1 (19.2)	92.1 (49.1)	93.0 (**50.7**)	**94.4** (27.7)	94.0 (37.2)	90.1 (22.1)	**94.4** (34.9)
Yellow light (11)	50.9 (1.4)	89.6 (11.1)	89.4 (30.5)	90.0 (29.9)	**98.6** (29.3)	94.6 (16.9)	97.6 (29.8)	96.2 (**39.0**)
Average	55.8 (14.5)	85.7 (32.3)	86.1 (38.9)	88.9 (39.0)	89.1 (32.33)	90.9 (38.7)	90.2 (37.2)	**93.3** (**47.9**)

Table 1. Accuracy and IOU (in parentheses) on our RGBD co-segmentation dataset. The first column shows the set name with the number of images. The top result for each set is highlighted in boldface.

constraints (*Our RGBD w/o M*), and without depth and mutex constraints (*Our RGB w/o M*). In the versions without mutex constraints, we simply set $\gamma = 0$ in Eq. (8). Two common performance metrics of image segmentation are used: accuracy, which is defined as the ratio of correctly labeled pixels in both the foreground and background, and intersection over union (IOU), which is the standard metric of PASCAL challenges.

Table 1 lists the accuracy and IOU scores of each method on our RGBD co-segmentation dataset. Some of the visual results are shown in Fig. 5. In [17], the discriminative clustering method is used to partition the images into foregrounds and backgrounds. On our dataset, we found that it often misses small objects, as shown in Fig. 5 (c), which leads to lower performance. The method in [19] uses a diffusion-based optimization framework that works well for object scale variations. However, its pixel-based segmentation was found to produce many meaningless fragments, as shown in Fig. 5 (d). The object-based method in [24] employs shortest path search to find the best candidates in multiple images. We found that it can produce results better than [17] and generally comparable in quality to [19]. Our method without depth obtains scores similar to [24] but is also able to handle noisy images where the common object appears more or less than once. With the depth cues, the RGBD co-segmentation methods, namely [24] + Depth and our RGBD method, are better than RGB methods in most of the cases. The depth cue helps to distinguish similarly-colored foregrounds and backgrounds ('Carving' set) and better deal with object size/viewpoint changes ('Soda can' set) and illumination variation ('Green bowl' set). For [24], its directed graph connects only adjacent nodes, and this lack of global constraints can lead to a local solution. For example, in the 'Red cap' set, the complex background misleads the co-segmentation of [24] + Depth. By contrast, our method based on a fully-connected graph is more robust to complex backgrounds and object occlusion. Without mutex constraints, our method will output multiple candidates for one image, and we take the union of these candidates as the segmentation result. This generally leads to worse performance, as shown in Table 1.

Methods	[17]	[32]	[35]	[31]	[36]	Our RGB	Our RGBD
Alaska bear	74.8	86.4	90.0	90.0	90.4	92.8	**93.5**
Baseball	73.0	90.5	90.9	90.9	94.2	93.1	**96.5**
Stonehenge 1	56.6	87.3	63.3	91.3	92.5	86.7	**93.0**
Stonehenge 2	86.0	88.4	**88.8**	84.2	87.2	75.7	83.5
Soccer	76.4	82.6	87.5	86.7	89.4	**93.0**	92.1
Ferrari	85.0	84.3	89.9	92.7	**95.6**	83.5	91.7
Taj Mahal	73.7	88.7	91.1	81.7	**92.6**	84.9	88.7
Elephant	70.1	75.0	43.1	86.2	86.7	90.1	**90.4**
Panda	84.0	60.0	**92.7**	92.2	88.6	80.4	81.2
Kite	87.0	89.8	90.3	94.9	93.9	93.7	**96.6**
Kite panda	73.2	78.3	90.2	90.9	**93.1**	77.1	83.8
Gymnastics	90.9	87.1	91.7	**97.7**	90.4	95.8	95.4
Skating	**82.1**	76.8	77.5	79.9	78.7	81.7	81.7
Balloon	85.2	89.0	90.1	92.7	90.4	92.5	**96.5**
Statue	90.6	91.6	93.8	91.1	**96.8**	86.0	92.7
Bear	74.0	80.4	**95.3**	86.2	88.1	85.3	94.8
Average	78.9	83.5	85.4	89.6	90.5	86.8	**90.7**

Table 2. Accuracy scores on the iCoseg dataset.

4.2. Application to RGB images with estimated depth maps

We have also applied our RGBD co-segmentation method to RGB images with depth maps that have been estimated through non-parametric depth sampling [18]. In this experiment, we evaluate our method on the iCoseg co-segmentation dataset [2], which is the largest publicly available co-segmentation benchmark. We used the original implementation of [18] with default parameters to estimate the depth map for each image in the iCoseg dataset. For this dataset, since the estimated depth maps by [18] are coarse and sometimes inaccurate, we employ the 2D proposal method in [11] to generate the candidates. Our al-

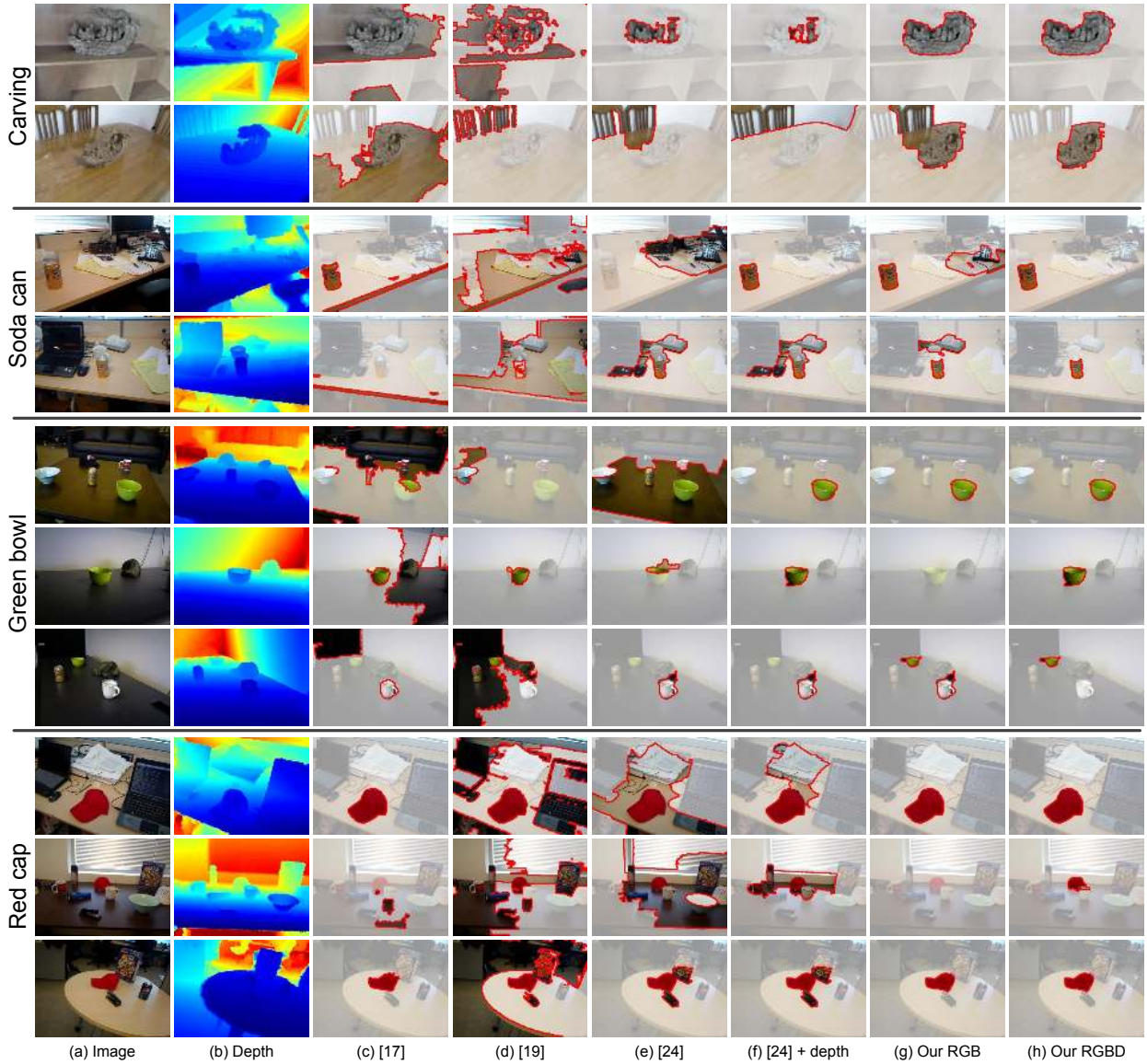

Figure 5. Some co-segmentation results on our RGBD co-segmentation dataset. From left to right: input images, depth maps, results of [17], [19], [24], [24] + depth, our RGB, and our RGBD co-segmentation. (Best viewed in color.)

gorithm is compared to the state-of-the-art co-segmentation methods [17, 35, 32, 31, 36]. Since the code for [35, 32, 36] has not been made available, we directly report the accuracy scores provided in these papers. Table 2 lists the accuracy scores on the iCoseg dataset, with the same number of images used in all of the methods.

Our method without depth obtains a slight improvement over the other object-based co-segmentation method [35] and also outperforms the methods in [17, 32]. With the estimated depth maps, our RGBD co-segmentation exhibits significant improvement in most of the cases. Fig. 6 displays some results of our RGB and RGBD co-segmentation on the iCoseg dataset. The estimated depth maps, though coarse and sometimes inaccurate, can nevertheless help to distinguish the common objects from the backgrounds in some cases, e.g., the 'Bear' and 'Taj Mahal' sets, where the depth maps show clear boundaries between object and background. Some estimated depth maps are highly inaccurate but still provide relative depth information between the object and background that is useful for separating them, e.g., the 'Stonehenge' and 'Elephant' sets. Moreover, this experiment also demonstrates the proper handling of the multiple instances of the common foreground, e.g., the 'Soccer' set. However, our method with estimated depth maps does not significantly outperform the state-of-the-art co-segmentation method [36], which is based on consisten-

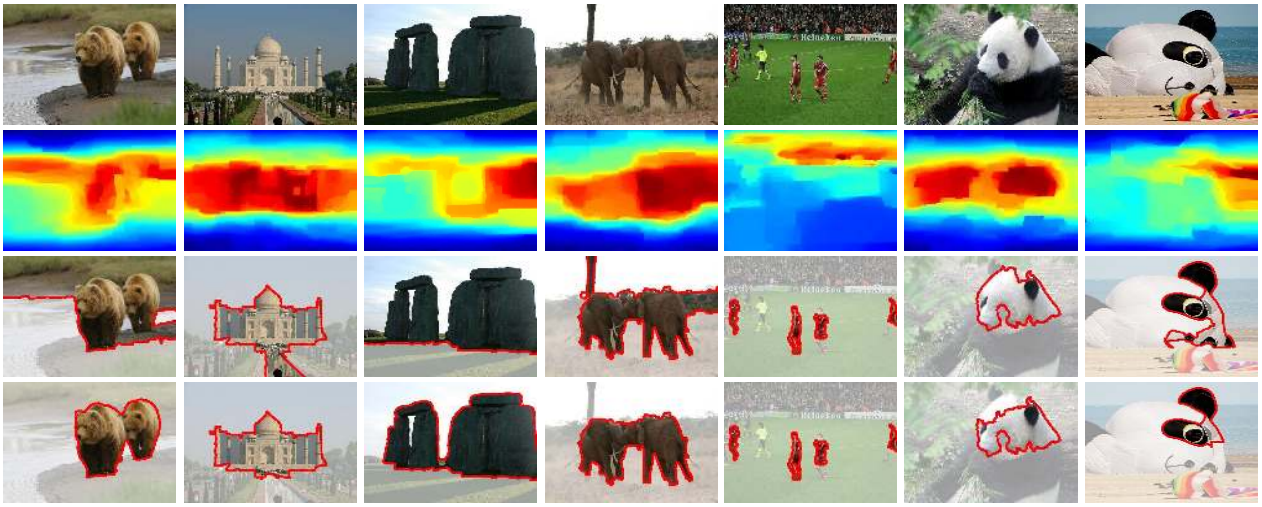

Figure 6. Our RGBD co-segmentation results on the iCoseg dataset. Top to bottom: one of the input images, its estimated depth map, and the results of our RGB and our RGBD co-segmentation. (Best viewed in color.)

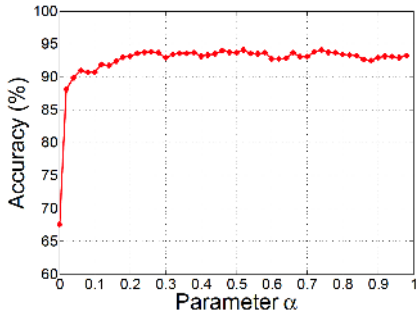

Figure 7. The average accuracy scores for different values of pairwise weight α on our RGBD co-segmentation dataset.

t functional maps for transporting properties between the RGB images. One possible explanation is that our method employs object proposals as the basic element of processing, which may fail to find the entire region when the object is composed of multiple highly diverse components, e.g., the image sets of 'Panda' and 'Kite panda' in the last two columns of Fig. 6.

4.3. Discussion

Parameter evaluation: Our method contains two parameters: pairwise weight α and mutex weight γ in Eq. (8). The mutex weight γ determines whether the solution from Eq. (7) will satisfy the mutex constraints. Thus, γ is set to a sufficiently large value and has little effect on co-segmentation results. We take the accuracy as an example to analyze the sensitivity of α, as shown in Fig. 7, where we fix $\gamma = 100$ and change the pairwise weight α. It can be seen that the performance is fairly stable for different values of the pairwise weight, when α is larger than 0.2. Note that when $\alpha = 0$, only the unary term is active, which results in selecting all the candidates except those excluded by the mutex constraint.

	Accuracy	IOU
[17]	55.7 ± 0.6	13.0 ± 0.6
[19]	81.5 ± 0.6	22.1 ± 1.8
[24]	85.3 ± 1.1	31.7 ± 0.9
[24] + Depth	88.7 ± 0.4	30.7 ± 0.6
Our RGB	87.1 ± 0.8	31.7 ± 1.0
Our RGBD	**92.3** ± 0.6	**38.8** ± 2.8

Table 3. Average accuracy and IOU on the RGBD co-segmentation dataset with noisy images.

Images sets with noisy images: We also conducted an experiment specifically for image sets which include noisy images that are missing the common foreground object. Here, we employ our RGBD co-segmentation dataset but add two random unrelated images to each image set. We compare our method with other methods [17, 19, 24]. We repeat the experiment five times with different random outlier images. Table 3 shows the average performance scores, which indicate that our method more effectively handles cases of noisy images.

5. Conclusion

We have proposed an object-based co-segmentation method for RGBD images that makes effective use of depth information. With a fully-connected graph structure and mutex constraints, our method is able to properly deal with image sets that contain noisy images with more than or less than one common foreground object.

Acknowledgements: This work is supported by the Singapore A*STAR SERC Grant (112-148-0003).

References

[1] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *CVPR*, pages 328–335, 2014.

[2] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. Interactively co-segmentating topically related images with intelligent scribble guidance. *IJCV*, 93(3):273–292, 2011.

[3] L. Bo, X. Ren, and D. Fox. Depth kernel descriptors for object recognition. In *IROS*, pages 821–826, 2011.

[4] W. Brendel and S. Todorovic. Segmentation as maximum-weight independent set. In *NIPS*, 2010.

[5] X. Cao, Y. Cheng, Z. Tao, and H. Fu. Co-saliency detection via base reconstruction. In *ACM MM*, pages 997–1000, 2014.

[6] X. Cao, Z. Tao, B. Zhang, H. Fu, and W. Feng. Self-adaptively weighted co-saliency detection via rank constraint. *TIP*, 23(9):4175–4186, 2014.

[7] J. Carreira and C. Sminchisescu. CPMC: Automatic object segmentation using constrained parametric min-cuts. *TPAMI*, 34(7):1312–1328, 2012.

[8] K. Chang, T. Liu, and S. Lai. From co-saliency to co-segmentation: An efficient and fully unsupervised energy minimization model. In *ICCV*, pages 2129–2136, 2011.

[9] Y. Cheng, H. Fu, X. Wei, J. Xiao, and X. Cao. Depth enhanced saliency detection method. In *Proceedings of International Conference on Internet Multimedia Computing and Service*, pages 23–28, 2014.

[10] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

[11] I. Endres and D. Hoiem. Category-independent object proposals with diverse ranking. *TPAMI*, 36(2):222–234, 2014.

[12] H. Fu, X. Cao, and Z. Tu. Cluster-based co-saliency detection. *TIP*, 22(10):3766–3778, 2013.

[13] H. Fu, D. Xu, B. Zhang, and S. Lin. Object-based multiple foreground video co-segmentation. In *CVPR*, pages 3166–3173, 2014.

[14] S. Gupta, R. Girshick, and P. Arbel. Learning Rich Features from RGB-D Images for Object Detection and Segmentation. In *ECCV*, pages 345–360, 2014.

[15] D. Hochbaum and V. Singh. An efficient algorithm for co-segmentation. In *CVPR*, pages 269–276, 2009.

[16] A. Humayun and J. Rehg. RIGOR : Reusing Inference in Graph Cuts for generating Object Regions. In *CVPR*, pages 336–343, 2014.

[17] A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, pages 1943–1950, 2010.

[18] K. Karsch, C. Liu, and S. Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *TPAMI*, 36(11):2144–2158, 2014.

[19] G. Kim, E. Xing, L. Fei-Fei, and T. Kanade. Distributed cosegmentation via submodular optimization on anisotropic diffusion. In *ICCV*, pages 169–176, 2011.

[20] P. Krahenbuhl and V. Koltun. Geodesic object proposals. In *ECCV*, pages 725–739, 2014.

[21] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view RGB-D object dataset. In *ICRA*, pages 1817–1824, 2011.

[22] C. Lang, T. Nguyen, H. Katti, K. Yadati, M. Kankanhalli, and S. Yan. Depth matters: Influence of depth cues on visual saliency. In *ECCV*, pages 101–115, 2012.

[23] T. Ma and L. Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *CVPR*, pages 670–677, 2012.

[24] F. Meng, H. Li, G. Liu, and K. Ngan. Object co-segmentation based on shortest path algorithm and saliency model. *TMM*, 14(5):1429–1441, 2012.

[25] A. Mishra, A. Shrivastava, and Y. Aloimonos. Segmenting "simple" objects using RGB-D. In *ICRA*, pages 4406–4413, 2012.

[26] L. Mukherjee, V. Singh, and J. Peng. Scale invariant cosegmentation for image groups. In *CVPR*, pages 1881–1888, 2011.

[27] Y. Niu, Y. Geng, X. Li, and F. Liu. Leveraging stereopsis for saliency analysis. In *CVPR*, pages 454–461, 2012.

[28] H. Peng, B. Li, W. Xiong, W. Hu, and R. Ji. RGBD Salient Object Detection: A Benchmark and Algorithms. In *ECCV*, pages 92–109, 2014.

[29] A. Richtsfeld, M. Thomas, J. Prankl, M. Zillich, and M. Vincze. Segmentation of unknown objects in indoor environments. In *IROS*, pages 4791–4796, 2012.

[30] C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of Image Pairs by Histogram Matching - Incorporating a Global Constraint into MRFs. In *CVPR*, pages 993–1000, 2006.

[31] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu. Unsupervised joint object discovery and segmentation in internet images. In *CVPR*, pages 1939–1946, 2013.

[32] J. Rubio, J. Serrat, A. Lopez, and N. Paragios. Unsupervised co-segmentation through region matching. In *CVPR*, pages 749–756, 2012.

[33] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from RGBD images. In *ECCV*, pages 746–760, 2012.

[34] J. Uijlings, K. Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013.

[35] S. Vicente, C. Rother, and V. Kolmogorov. Object cosegmentation. In *CVPR*, pages 2217–2224, 2011.

[36] F. Wang, Q. Huang, and L. J. Guibas. Image co-segmentation via consistent functional maps. In *CVPR*, pages 849–856, 2013.

[37] C. Yang, L. Zhang, H. Lu, X. Ruan, and M. Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, pages 3166–3173, 2013.

[38] W. Zhu, S. Liang, Y. Wei, and J. Sun. Saliency optimization from robust background detection. In *CVPR*, pages 2814–2821, 2014.