POD: Discovering Primary Objects in Videos Based on Evolutionary Refinement of Object Recurrence, Background, and Primary Object Models

Yeong Jun Koh
Korea University
yjkoh@mcl.korea.ac.kr

Won-Dong Jang
Korea University
wdjang@mcl.korea.ac.kr

Chang-Su Kim
Korea University
changsukim@korea.ac.kr

Abstract

A primary object discovery (POD) algorithm for a video sequence is proposed in this work, which is capable of discovering a primary object, as well as identifying noisy frames that do not contain the object. First, we generate object proposals for each frame. Then, we bisect each proposal into foreground and background regions, and extract features from each region. By superposing the foreground and background features, we build the object recurrence model, the background model, and the primary object model. We develop an iterative scheme to refine each model evolutionarily using the information in the other models. Finally, using the evolved primary object model, we select candidate proposals and locate the bounding box of a primary object by merging the proposals selectively. Experimental results on a challenging dataset demonstrate that the proposed POD algorithm extracts primary objects accurately and robustly.

1. Introduction

Discovering a primary object is an essential task in computer vision, since a repeatedly appearing object in multiple images or videos conveys useful information about those signals. For example, the segmentation of a common object across frames in a video facilitates the video summarization. Also, object discovery techniques can be used for collecting objects of the common class from many images to train an object detector. Without those techniques, the collection would demand a lot of human efforts. Moreover, in a content-based image retrieval system, object discovery techniques can identify noisy frames, which do not contain a target object, and exclude them from the system.

Many techniques have been developed to discover a primary object. They can be classified into three categories: object discovery, cosegmentation, and video object segmentation. In object discovery [6, 7, 13, 23, 26, 29, 33], objects of the common class are localized in a set of images or videos. In cosegmentation [11, 12, 15, 19, 25, 27, 31], assuming that an identical object appears in multiple images, the object is delineated at the pixel-level in each image. In video object segmentation, objects are separated from its background [9,10,16–18,22,30,34–36], or dense object segments are determined based on motion clustering [4,21,28]. However, the primary object discovery (POD) is still a challenging problem due to a wide variety of difficulties, such as cluttered and diverse backgrounds, object appearance variations, interrupting objects, and noisy frames.

In this work, we propose a novel POD algorithm, which discovers an identical object in a video sequence. The proposed algorithm has the following main advantages:

- It discovers a primary object efficiently in a single video, whereas most conventional object discovery techniques assume a large set of images or videos.
- It provides robust performance even when a primary object exhibits abrupt motions or is interfered by other moving objects. This is because the proposed algorithm does not depend on motion information.
- While discovering a primary object, it also identifies noisy frames that do not contain the object.

To this end, we first generate object proposals for each frame. Then, we divide each proposal into foreground and background regions and extract superpixel-based features from each region. By superposing the foreground and background features, we construct three models: object recurrence, background, and primary object models. We iteratively update each model by exploiting the other models. During the iteration, we also detect noisy frames. Finally, we choose candidate proposals using the primary object model and discover a primary object by merging the proposals selectively. Experimental results on an extensive dataset demonstrate that the proposed POD algorithm discovers primary objects effectively and robustly.

The rest of this paper is organized as follows: Section 2 reviews conventional techniques, related to POD. Section 3 describes the proposed POD algorithm. Section 4 discusses experimental results. Section 5 concludes this work.

2. Related Work

2.1. Object Discovery

Object discovery or co-localization is a process to find objects of the same class over multiple images or videos. To achieve this goal over videos, Prest *et al*. [23] extracted spatiotemporal tubes based on the motion segmentation [4], and jointly selected one tube in each video. Tang *et al*. [29] proposed the image-box strategy to determine the common object and to identify noisy images without the object. Joulin *et al*. [13] selected object proposals, containing the common object, by employing the Frank-Wolfe algorithm. Even from an image pool containing multi-class objects, Cho *et al*. [7] discovered an object by combining the part-based region matching with the foreground localization.

To achieve pixel-level object co-localization, several algorithms have been proposed to perform object discovery and segmentation simultaneously. Rubinstein *et al*. [26] developed a saliency-driven object discovery algorithm using pixel correspondences between images. Chen *et al*. [6] divided a set of images into subcategories, and then trained the object model and detector for each subcategory to delineate objects. Wang *et al*. [33] proposed an energy minimization scheme for simultaneous object discovery and segmentation.

These object discovery techniques collect objects of the same class, and thus are useful for training an object detector. However, since most discovery techniques [6, 7, 13, 23, 26, 29, 33] require a large dataset of images or videos, they are unsuitable for finding a primary object in a single video.

2.2. Cosegmentation

The objective of cosegmentation is to discover an identical object within a set of images. However, note that sometimes cosegmentation techniques are used for pixel-level object co-localization, *i.e*., for delineating objects of the same class, instead of the same object. Rother *et al*. [25] first proposed a cosegmentation algorithm using a generative Markov random field (MRF) model. Instead of the generative model, Mukherjee *et al*. [19] built a successive model to constrain foreground histograms to be similar to one another. To delineate an identical object, Hochbaum and Singh [11] optimized an MRF model and maximized the similarity between the foregrounds simultaneously.

Joulin *et al*. [12] applied a discriminative clustering technique to the cosegmentation problem. Chang *et al*. [5] introduced a co-saliency prior to locate the common object. Inspired by the anisotropic heat diffusion, Kim *et al*. [14] proposed a scalable cosegmentation algorithm. Vicente *et al*. [31] determined similar object proposals from multiple images by learning a classifier. To match objects among images, Rubio *et al*. [27] exploited a region matching technique, and solved pixel-level and region-level energy minimization problems. Wang *et al*. [32] extracted cyclic functional maps and generated segmentation functions to indicate the foreground probability of each superpixel. Lee *et al*. [15] proposed the notion of multiple random walkers on a graph, and applied it to the image cosegmentation using the repulsive restart rule.

Although the cosegmentation techniques [5, 11, 12, 14, 15, 19, 25, 27, 31, 32] can delineate the same object in a set of images, they are not effective for segmenting objects in a video since they do not consider noisy frames.

2.3. Video Object Segmentation

In video object segmentation, a primary object in a video sequence is separated from its background. Shi and Malik [28] constructed a spatiotemporal graph and partitioned it using the normalized cuts. Brox and Malik [4] traced point trajectories and clustered them. Ochs and Brox [20] converted clusters of sparse trajectories into dense object segments using a diffusion process. Ochs and Brox [21] also applied the spectral clustering to a hypergraph representing point trajectories. However, these motion segmentation algorithms [4, 20, 21, 28] do not provide the priority of a segment and thus cannot identify a primary object.

Lee *et al*. [16] delineated a key object by selecting a hypothesis, composed of object proposals across frames. Ma and Latecki [18] extracted a primary object, by determining a maximum weight clique in a graph of object proposals. Zhao *et al*. [36] discovered objects of interest in a video by employing the latent Dirichlet allocation [3]. Li *et al*. [17] formed segment tracks from a pool of figure-ground segments, and refined the segmentation using the composite statistical inference. Zhang *et al*. [35] introduced a layered graph of object proposals, and selected a node for each frame using dynamic programming. Papazoglou and Ferrari [22] estimated motion boundaries to discover moving objects. Faktor *et al*. [9] presented the non-local consensus voting scheme, which used saliency maps as initial likelihood to find primary objects. Wang *et al*. [34] detected saliency maps using geodesic distances to discover a salient object. Giordano *et al*. [10] performed video object segmentation by exploiting the continuity of superpixels in consecutive frames. Taylor *et al*. [30] delineated objects in a video, by identifying occluders and determining occlusion relations. However, most of these algorithms [9, 10, 16–18, 22, 30, 34–36] assume that a primary object has distinct motions from the background. Moreover, they cannot identify an primary object across different shots, when those shots have diverse scene contents and contain different non-primary objects.

3. Proposed Algorithm

Our goal is to discover the bounding boxes that trace a primary object in a sequence of video frames $\mathcal{V}$ =

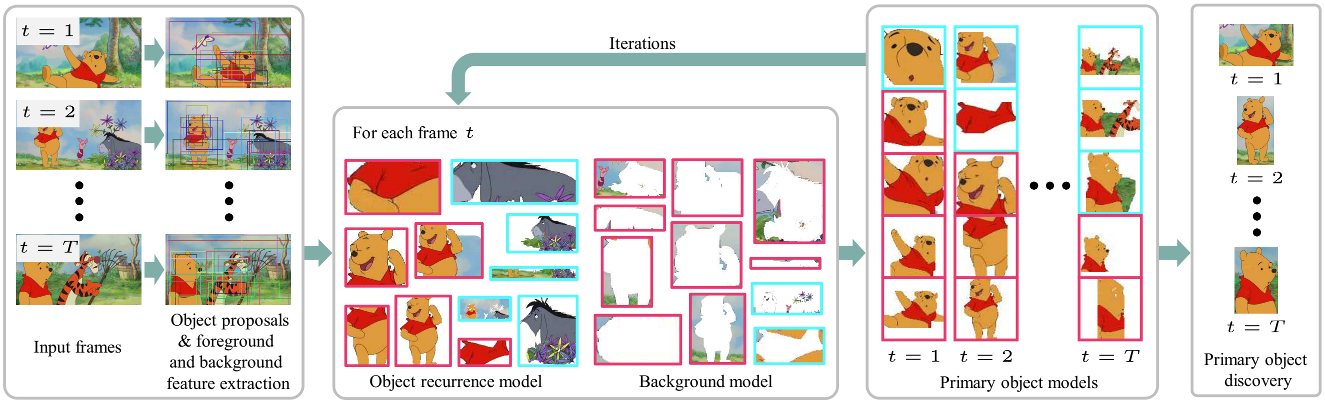


Figure 1. An overview of the proposed POD algorithm. In the evolutionary refinement of the three models, high and low weights are indicated by coloured boundaries □ and □, respectively.

$\{I_1, \ldots, I_T\}$. We assume that a primary object appears in most frames, but it need not be in all frames. Hence, while detecting a primary object, we also identify noisy frames that do not contain the object.

Figure 1 shows an overview of the proposed POD algorithm. First, we generate object proposals for each frame and extract foreground and background features from each proposal. Second, for each frame, we build the object recurrence model and the background model using those features. Third, using the two base models, we combine the foreground features linearly to construct the primary object model, which is finally used to discover a primary object. Note that the three models are iteratively constructed.

3.1. Modelling a Primary Object

Object Proposal Generation: For each frame, we obtain a set of object proposals by employing the Alexe *et al.*'s algorithm [2]. Let $\mathcal{O}_t = \{o_{t,1}, o_{t,2}, \ldots, o_{t,m}\}$ be the set of proposals at frame t, where m is the number of proposals that is set to 20 in this work. From each proposal $o_{t,i}$, we extract the foreground feature $\mathbf{p}_{t,i}$ and the background feature $\mathbf{q}_{t,i}$. Specifically, we first divide each proposal into foreground and background regions using the GrabCut algorithm [24], which was designed to bisect a manually annotated boxed region. Note that a proposal (or the corresponding box) is automatically generated by [2], and thus the manual annotation is not performed in this work. Then, we extract the foreground and background features, $\mathbf{p}_{t,i}$ and $\mathbf{q}_{t,i}$, from the foreground and background regions, respectively. For the feature representation, we adopt the bag-of-visual-words approach. Given a video sequence, we over-segment each frame into 1,000 superpixels using the SLIC algorithm [1], and then encode the average LAB colors of all superpixels into 100 codewords. Then, the foreground feature $\mathbf{p}_{t,i}$ is obtained by recording the histogram of the codewords for the foreground region, and the background feature $\mathbf{q}_{t,i}$ is obtained in a similar manner. Both $\mathbf{p}_{t,i}$ and $\mathbf{q}_{t,i}$ are normalized, and thus they can be regarded as probability distributions.

Object Recurrence Model: A primary object occurs repeatedly in a video sequence. This recurrence property implies that some of the object proposals contain a whole or part of the primary object. Thus, by mixing the foreground features of the proposals, we can approximate the features of the primary object. Based on this observation, we define the object recurrence model $\mathbf{R}_t^{(\theta)}(\Gamma_t)$ at frame t at iteration θ as

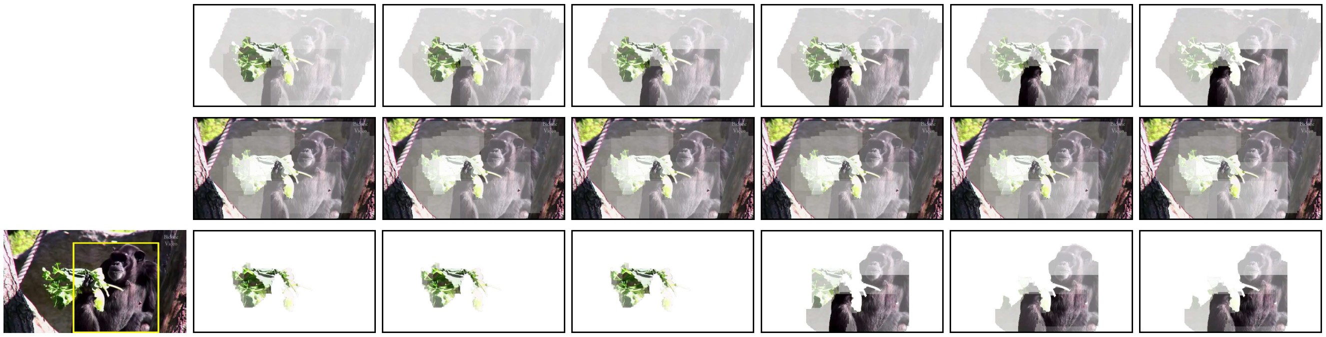
$$\mathbf{R}_t^{(\theta)}(\Gamma_t) = \sum_{i=1}^{m} \gamma_{t,i}\mathbf{p}_{t,i}. \tag{1}$$

where $\Gamma_t = (\gamma_{t,1}, \ldots, \gamma_{t,m})$ denotes the set of the recurrence weights such that $0 \le \gamma_{t,i} \le 1$ and $\sum_i \gamma_{t,i} = 1$. Each recurrence weight $\gamma_{t,i}$ indicates the likelihood that the corresponding feature $\mathbf{p}_{t,i}$ comes from the primary object. Notice that, in the object recurrence model, only the foreground features $\mathbf{p}_{t,i}$ are used and the background features $\mathbf{q}_{t,i}$ are not considered.

To obtain Γ_t, we use the primary object model for each frame, which will be discussed later in this section. The primary object model is a refinement of the object recurrence model by using both foreground and background features. In this work, we update the object recurrence models and the primary object models iteratively. Let $\mathcal{P}^{(\theta)} = (\mathbf{P}_1^{(\theta)}, \ldots, \mathbf{P}_T^{(\theta)})$ denote the set of the primary object models at iteration θ. At the start of iterations, the primary object model $\mathbf{P}_t^{(0)}$ at frame t is initialized to the average of the foreground features, given by

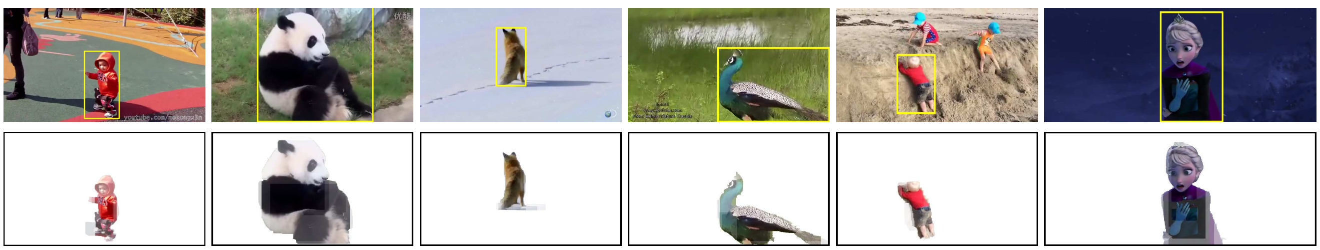
$$\mathbf{P}_t^{(0)} = \frac{1}{m}\sum_{i=1}^{m} \mathbf{p}_{t,i}. \tag{2}$$

Given the primary object models $\mathcal{P}^{(\theta-1)}$ at the previous iteration $\theta - 1$, we compute the recurrence weights Γ_t to make the object recurrence model $\mathbf{R}_t^{(\theta)}(\Gamma_t)$ approximate the feature of the primary object. To this end, we define



(a) Frame t (b) $\theta = 1$ (c) $\theta = 2$ (d) $\theta = 3$ (e) $\theta = 4$ (f) $\theta = 5$ (g) $\theta = 6$

Figure 2. Iterative primary object modeling on the "Monkey" sequence. An input frame and its ground-truth box at frame t are shown in (a). At each iteration θ in (b)$\sim$(g), from top to bottom, the object recurrence model $\mathbf{R}_t^{(\theta)}(\Gamma_t^*)$, the background model $\mathbf{B}_t^{(\theta)}$, and the primary object model $\mathbf{P}_t^{(\theta)}$ are shown. To visualize $\mathbf{R}_t^{(\theta)}(\Gamma_t^*)$ and $\mathbf{P}_t^{(\theta)}$, the foreground regions of proposals are linearly superposed using weights $\gamma_{t,i}^*$ and $\omega_{t,i}$, respectively. Similarly, for $\mathbf{B}_t^{(\theta)}$, the background regions are superposed using weights $\lambda_{t,i}$.



(a) Baby (b) Babypanda (c) Fox (d) Peacock (e) Baby2 (f) Frozen

Figure 3. More examples of primary object models after the convergence. The top row shows ground-truth bounding boxes, and the bottom row are the corresponding primary object models.

the global primary object model $\overline{\mathbf{P}}^{(\theta-1)}$ as

$$\overline{\mathbf{P}}^{(\theta-1)} = \frac{\sum_{t=1}^{T} c_t^{(\theta-1)} \mathbf{P}_t^{(\theta-1)}}{\sum_{t=1}^{T} c_t^{(\theta-1)}} \tag{3}$$

where the binary indicator $c_t^{(\theta-1)}$ is 0 if frame t is detected as noisy at the previous iteration $\theta - 1$, and 1 otherwise. Thus, $\overline{\mathbf{P}}^{(\theta-1)}$ is averaged over only the frames containing the primary object. Note that, if we construct the recurrence model at frame t using the primary object model at the corresponding frame only, each recurrence model may describe a different object. We hence use the global model to make all recurrence models represent an identical object.

Consequently, the object recurrence model $\mathbf{R}_t^{(\theta)}(\Gamma_t)$ should be similar to the global primary object model $\overline{\mathbf{P}}^{(\theta-1)}$. We adopt the Kullback-Leibler divergence to measure the dissimilarity between $\mathbf{R}_t^{(\theta)}(\Gamma_t)$ and $\overline{\mathbf{P}}^{(\theta-1)}$,

$$D(\mathbf{R}_t^{(\theta)}(\Gamma_t) || \overline{\mathbf{P}}^{(\theta-1)}). \tag{4}$$

Note that the Kullback-Leibler divergence or relative entropy

$$D(\mathbf{u}||\mathbf{v}) = \sum_i u_i \log \frac{u_i}{v_i} \tag{5}$$

is often used to measure the distance between two probability distributions $\mathbf{u}$ and $\mathbf{v}$, and it is convex in terms of both $\mathbf{u}$ and $\mathbf{v}$ [8]. We then estimate the optimal set of the recurrence weights Γ_t^* by

$$\Gamma_t^* = \arg\min_{\Gamma_t} D(\mathbf{R}_t^{(\theta)}(\Gamma_t) || \overline{\mathbf{P}}^{(\theta-1)}) \tag{6}$$

subject to

$$0 \le \gamma_{t,i} \le 1, \quad \sum_i \gamma_{t,i} = 1. \tag{7}$$

Because of the convexity of relative entropy [8], the constrained optimization in (6) is a convex optimization problem, which can be easily solved. Finally, using the optimal Γ_t^*, we obtain the object recurrence model $\mathbf{R}_t^{(\theta)}(\Gamma_t^*)$ at frame t. Note that the optimal Γ_t^* makes the object recurrence model $\mathbf{R}_t^{(\theta)}(\Gamma_t^*)$ as close to the global primary object model $\overline{\mathbf{P}}^{(\theta-1)}$ as possible.

Background Model: Next, we define the background model $\mathbf{B}_t^{(\theta)}$ at frame t at iteration θ using the background features of the proposals as

$$\mathbf{B}_t^{(\theta)} = \frac{\sum_{i=1}^{m} \lambda_{t,i} \mathbf{q}_{t,i}}{\sum_{i=1}^{m} \lambda_{t,i}} \tag{8}$$

where $\lambda_{t,i}$ is the weight for the background feature $\mathbf{q}_{t,i}$ of the ith proposal. The background model should be distinguishable from the object recurrence model. Therefore, we assign a higher weight $\lambda_{t,i}$, when the feature $\mathbf{q}_{t,i}$ is more dissimilar from $\mathbf{R}_t^{(\theta)}(\Gamma_t^*)$. In other words, we set the weight $\lambda_{t,i}$ as

$$\lambda_{t,i} = d_\chi(\mathbf{q}_{t,i}, \mathbf{R}_t^{(\theta)}(\Gamma_t^*)) \tag{9}$$

where $d_\chi(\cdot,\cdot)$ denotes the chi-square distance.

Primary Object Model: Even though the object recurrence model $\mathbf{R}_t^{(\theta)}(\Gamma_t^*)$ in (1) and (6) roughly represents the feature of the primary object, it does not fully exploit the background information in (8). Therefore, we attempt to obtain a more refined model of the primary object. We define the primary object weight $\omega_{t,i}$ for the foreground feature $\mathbf{p}_{t,i}$ of the ith proposal as

$$\omega_{t,i} = \frac{d_\chi(\mathbf{p}_{t,i}, \mathbf{B}_t^{(\theta)})}{d_\chi(\mathbf{p}_{t,i}, \mathbf{R}_t^{(\theta)}(\Gamma_t^*))}. \tag{10}$$

A high $\omega_{t,i}$ indicates that the foreground feature $\mathbf{p}_{t,i}$ is similar to the object recurrence model but dissimilar from the background model.

We select the top-5 proposals according to the primary object weights. Then, we determine the primary object model $\mathbf{P}_t^{(\theta)}$ at frame t at the current iteration θ, by superposing the foreground features of the selected proposals,

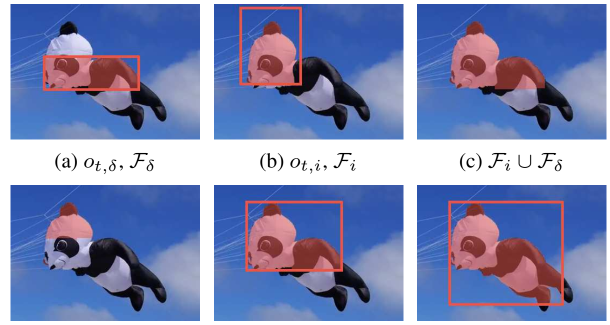
$$\mathbf{P}_t^{(\theta)} = \frac{\sum_{i\in\mathcal{I}_t}\omega_{t,i}\mathbf{p}_{t,i}}{\sum_{i\in\mathcal{I}_t}\omega_{t,i}} \tag{11}$$

where $\mathcal{I}_t$ is the index set of the top-5 proposals at frame t. Note that these primary object models $\mathcal{P}^{(\theta)} = (\mathbf{P}_1^{(\theta)}, \ldots, \mathbf{P}_T^{(\theta)})$ are, in turn, used to update the object recurrence models in (1) and (6) at the next iteration $\theta + 1$.

Noisy Frame Detection: After obtaining the primary object models for all frames, we compute the distance between the global model $\overline{\mathbf{P}}^{(\theta-1)}$ in (3) and each model $\mathbf{P}_t^{(\theta)}$ in (11). We declare frame t as noisy if $d_\chi(\mathbf{P}_t^{(\theta)}, \overline{\mathbf{P}}^{(\theta-1)})$ is larger than a threshold 0.7. This information is recorded in the indicator vector $\mathbf{c}^{(\theta)} = (c_1^{(\theta)}, \ldots, c_T^{(\theta)})$. Here, $c_t^{(\theta)}$ is 0 if frame t is noisy, and 1 otherwise.

Iterative Modelling: We update the object recurrence models, the background models, and the primary object models iteratively, until $d(\overline{\mathbf{P}}^{(\theta)}, \overline{\mathbf{P}}^{(\theta-1)})$ converges to zero.

Figure 2 illustrates how the three models evolve as the iteration goes on. Initially, at $\theta = 1$, the primary object model expresses the green features of a vegetable. However, after the convergence at $\theta = 6$, the primary object model faithfully represents the features of the primary object, *i.e.*, the monkey. For most sequences, the proposed algorithm converges after 5 to 10 iterations. Figure 3 shows examples of primary object models after the convergence.



(a) $o_{t,\delta}, \mathcal{F}_\delta$ (b) $o_{t,i}, \mathcal{F}_i$ (c) $\mathcal{F}_i \cup \mathcal{F}_\delta$

(d) $\mathcal{F}_i \setminus \mathcal{F}_\delta$ (e) Merged box (f) Final box

Figure 4. An example of the primary object discovery. Since a candidate proposal $o_{t,i}$ yields a high score $\Psi(o_{t,i}, o_{t,\delta})$, we merge $o_{t,i}$ to $o_{t,\delta}$ and update the box in (e). After merging with other proposals, we obtain the final box in (f). Coloured boundaries and regions depict bounding boxes and foreground regions, respectively.

3.2. Discovering a Primary Object

After the convergence of the global primary object model $\overline{\mathbf{P}} = \overline{\mathbf{P}}^{(\theta-1)} = \overline{\mathbf{P}}^{(\theta)}$ in Section 3.1, we discover the primary object through the video sequence.

For each frame t, we have the index set $\mathcal{I}_t$ of the top-5 proposals in (11). Among those proposals, we choose the main proposal that has the minimum distance from the global model $\overline{\mathbf{P}}$, whose index is given by

$$\delta = \arg\min_{i\in\mathcal{I}_t} d_\chi(\mathbf{p}_{t,i}, \overline{\mathbf{P}}). \tag{12}$$

We then merge the main proposal $o_{t,\delta}$ with each candidate proposal $o_{t,i}$, where $i \in \mathcal{I}_t$ and $i \neq \delta$, by employing a score function $\Psi(o_{t,i}, o_{t,\delta})$.

Let $\mathcal{F}_\delta$ and $\mathcal{F}_i$ denote the foreground regions of a main proposal $o_{t,\delta}$ and a candidate proposal $o_{t,i}$, respectively, extracted by the GrabCut algorithm [24]. The boxes and the foreground regions of the main and candidate proposals are illustrated in Figure 4(a) and (b), respectively. We consider the union region $\mathcal{F}_i \cup \mathcal{F}_\delta$ in Figure 4(c) and the difference region $\mathcal{F}_i \setminus \mathcal{F}_\delta$ in Figure 4(d). Then, we extract the features $\mathbf{p}_{i\cup\delta}$ and $\mathbf{p}_{i\setminus\delta}$ from the union region and the difference region, respectively. Using these features, we evaluate the score function as

$$\Psi(o_{t,i}, o_{t,\delta}) = \frac{d_\chi(\mathbf{p}_{t,\delta}, \overline{\mathbf{P}})}{d_\chi(\mathbf{p}_{i\cup\delta}, \overline{\mathbf{P}})}\left(1 - d_\chi(\mathbf{p}_{i\setminus\delta}, \mathbf{p}_{t,\delta})\right). \tag{13}$$

Suppose that the foreground region $\mathcal{F}_i$ of the candidate proposal $o_{t,i}$ mostly covers the primary object and includes none or only a little part of the background. Then, the feature $\mathbf{p}_{i\cup\delta}$ of the union region should be similar to the global primary object model $\overline{\mathbf{P}}$, as the feature $\mathbf{p}_{t,\delta}$ of the main proposal is. In other words, the ratio $\frac{d_\chi(\mathbf{p}_{t,\delta},\overline{\mathbf{P}})}{d_\chi(\mathbf{p}_{i\cup\delta},\overline{\mathbf{P}})}$ should be close to 1. Moreover, the difference region should have a similar feature to the main proposal, and $d_\chi(\mathbf{p}_{i\setminus\delta}, \mathbf{p}_{t,\delta})$ should be

Algorithm 1 Primary Object Discovery (POD)

Input: A video sequence $\mathcal{V} = \{I_1, \ldots, I_T\}$

Generate object proposals for each frame
Divide each proposal into foreground and background regions
/* Modelling a primary object */ $\triangleright$ Sec. 3.1
repeat
  for each frame t **do**
    Compute the object recurrence model $\mathbf{R}_t^{(\theta)}(\Gamma_t^*)$ $\triangleright$ (1)
    Compute the background model $\mathbf{B}_t^{(\theta)}$ $\triangleright$ (8)
    Compute the primary object model $\mathbf{P}_t^{(\theta)}$ $\triangleright$ (11)
  end for
  Detect noisy frames and update $\mathbf{c}^{(\theta)}$
  Increase iteration index θ
until $d(\overline{\mathbf{P}}^{(\theta)}, \overline{\mathbf{P}}^{(\theta-1)})$ converges to zero
/* Discovering a primary object */ $\triangleright$ Sec. 3.2
for each frame t **do**
  Determine the main proposal $o_{t,\delta}$ $\triangleright$ (12)
  Merge the proposals selectively $\triangleright$ (13)
end for

Output: Bounding boxes of the primary object

small. Note that the chi-square distance ranges from 0 to 1. Thus, if $\Psi(o_{t,i}, o_{t,\delta})$ is larger than a threshold 0.4, we merge $o_{t,i}$ to the main proposal $o_{t,\delta}$. After the mergence, we put the bounding box that encloses the two foreground regions, as in Figure 4(e). This merging process is repeatedly performed with each candidate proposal $o_{t,i}$, where $i \in \mathcal{I}_t$ and $i \neq \delta$. Figure 4(f) shows that the final bounding box discovers the kite panda effectively.

Algorithm 1 summarizes the proposed POD algorithm.

4. Experimental Results

We test the proposed algorithm on the primary object video (POV) dataset and YouTube-Objects dataset [23]. As in the object discovery and localization techniques in [7, 13, 23, 29], we adopt the correct localization (CorLoc) metric, which measures the percentage of images correctly localized according to the PASCAL criterion: an estimated box $\mathcal{B}_{\mathrm{p}}$ is correct as compared with the ground-truth box $\mathcal{B}_{\mathrm{gt}}$, when the intersection over union (IoU) overlap ratio $\frac{|\mathcal{B}_{\mathrm{p}} \cap \mathcal{B}_{\mathrm{gt}}|}{|\mathcal{B}_{\mathrm{p}} \cup \mathcal{B}_{\mathrm{gt}}|}$ is larger than 0.5.

4.1. POV Dataset

We organize a dataset of 20 video sequences, whose durations vary from one to four minutes. Each video contains an identical primary object. In Table 1, we classify the 20 videos into four categories according to scene types and object motions: "Simple," "Static object," "Multi-objects," and "Animation." The "Simple" category consists of relatively easy videos, each of which contains a single object, few scene changes, and few noisy frames. In each "Static object" video, a primary object remains stationary in many frames without fast motions. Each "Multi-objects" video contains other objects in addition to a primary object. Each "Animation" video includes many noisy frames, has frequent scene changes and diverse backgrounds, and contains multiple objects. To reduce the amount of manual efforts for annotating ground truths, we sampled every 12th frame to obtain the temporally decimated sequences, and then annotated the bounding boxes for the sampled frames.

4.2. Comparative Performance Evaluation

In Table 1, we compare the proposed algorithm with the previous algorithms on cosegmentation [15], object discovery [7, 29], and video object segmentation [22, 35]. We obtain the results of the previous algorithms using the source codes, provided by the respective authors. Every attempt has been made to make a fair comparison. However, as detailed below, experimental conditions are slightly different according to the implementation issues of each algorithm.

Cosegmentation: We compare the proposed POD algorithm with the Lee *et al.*'s cosegmentation algorithm [15]. Note that [15] constructs a graph by connecting all superpixels in all frames. Thus, it requires a huge amount of memory that is proportional to the number of frames. To overcome this issue, we execute [15] on 10 randomly selected frames, repeat the test ten times, and report the average performance. [15] provides good performances on simple videos, which have temporally consistent backgrounds and no noisy frames. However, it is vulnerable to noisy frames or diverse backgrounds.

Object Discovery: The object discovery algorithms [7, 29] are tested under the same condition as the proposed algorithm. On all the video sequences, the proposed POD algorithm significantly outperforms these object discovery algorithms, which often produce large bounding boxes, containing background regions as well. This is because [7, 29] depend on the similarity of foreground proposals. However, background proposals also may be similar to one another, degrading the performances of these algorithm. Especially, the Cho *et al.*'s algorithm [7] assumes that background regions in different frames are dissimilar. However, this assumption is often invalid within a video sequence, and the background cannot be effectively discriminated from the foreground. Therefore, [7, 29] cannot deal with background clutters effectively for the video sequence in Figure 5(a).

Video Object Segmentation: We also compare the proposed POD algorithm with the video object segmentation algorithms in [22, 35]. Since these previous algorithms provide pixel-level segmentation results, we fit a bounding box to the largest connected component of a segmentation result, as done in [22]. Also, although the dataset is constructed by sampling every 12th frame, we triple the sampling rate and sample every 4th frame for the Papazoglou

Table 1. Performance comparison of the proposed algorithm with the conventional algorithms on the POV dataset using the CorLoc metric. The best results are boldfaced.

Category	Video (No. of frames)	Cosegmentation [15]	Object discovery [29]	Object discovery [7]	Video object segmentation [22]	Video object segmentation [35]	Proposed POD
Simple	Baby (528)	71.00	14.84	25.24	76.30	76.47	**82.96**
	Helicopter (200)	26.03	12.50	67.00	**77.50**	61.37	76.50
	Rabbit (459)	**93.31**	64.54	73.13	82.16	71.48	92.73
	Cat (299)	**93.00**	31.44	49.83	49.83	68.29	92.98
	Babypanda (361)	73.09	52.15	53.58	82.52	37.25	**94.18**
Static object	Yellow Bear (147)	3.47	27.59	40.00	22.76	85.52	**99.32**
	Raccoon (586)	49.25	38.83	45.02	27.32	70.88	**72.18**
	RC Car (287)	78.78	28.92	70.04	51.22	44.27	**88.50**
	Polarbear (247)	95.89	55.42	44.58	29.58	**99.17**	84.58
	Fox (393)	87.00	58.93	16.07	58.93	36.74	**88.30**
Multi-objects	Monkey (131)	58.00	30.53	36.64	29.77	45.80	**80.92**
	Panda (271)	53.72	42.74	41.91	21.16	49.80	**95.44**
	Car (322)	66.08	40.52	66.34	33.01	34.58	**66.99**
	Baby2 (349)	71.00	10.32	34.10	52.15	70.59	**87.97**
	Peacock (217)	64.14	24.17	45.02	48.34	51.89	**74.41**
	Dog (355)	20.56	17.39	10.44	19.42	40.29	**61.13**
Animation	Pooh (387)	42.00	14.21	3.88	6.98	5.67	**93.80**
	Yellow Larva (280)	4.00	33.57	6.43	26.07	21.43	**87.50**
	Frozen (434)	9.00	10.00	9.07	57.91	34.19	**64.88**
	Dooly (450)	54.00	11.33	16.22	32.22	30.67	**83.56**
Average		54.28	31.00	35.90	43.96	51.87	**83.44**

Table 2. Performance comparison of the proposed algorithm with the conventional algorithms [22, 23] on the YouTube-Objects dataset using the CorLoc metric. The best results are boldfaced.

	aeroplane	bird	boat	car	cat	cow	dog	horse	motorbike	train	avg
[23]	51.7	17.5	34.4	34.7	22.3	17.9	13.5	26.7	41.2	**25.0**	28.5
[22]	**65.4**	**67.3**	38.9	65.2	**46.3**	40.2	65.3	48.4	39.0	**25.0**	50.1
POD	64.3	63.2	**73.3**	**68.9**	44.4	**62.5**	**71.4**	**52.3**	**78.6**	23.1	**60.2**

and Ferrari's algorithm [22]. This is because [22] exploits the motion information and often benefits from a higher sampling rate. However, the CorLoc metric is applied to every 12th frame only, for which the ground truth is available. On the other hand, Zhang *et al.*'s algorithm [35] uses the same sampling rate as the proposed algorithm.

Except for the "Helicoptor" and "Polarbear" sequences, the proposed algorithm outperforms both the conventional algorithms [22, 35] significantly. For the "Simple" category, both algorithms provide sufficiently good performances. For the "Static object" category, the Papazoglou and Ferrari's algorithm [22] is vulnerable to static objects as shown in Figure 5(b), since it is highly dependent on motion information. For the "Multi-object" category, both algorithms [22, 35] often fail to identify primary objects, and thus detect non-primary objects. For example, another kite, instead of the panda kite, in Figure 5(c) and a vegetable in Figure 5(d) are regarded as primary objects due to their distinct colors and motions. In Figure 5(e), the conventional algorithms fail to locate the dog, which is occluded by obstacles, such as the fence. For the "Animation" category, the conventional algorithms suffer from noisy frames, frequent scene changes, and multi-objects. Therefore, they almost always fail to detect primary objects and identify noisy frames, as shown in Figure 5(f). In contrast, the proposed POD algorithm provides good performances on all categories, using the robust primary object models.

4.3. YouTube-Objects Dataset

YouTube-Objects [23] is a large dataset, containing videos for 10 object classes. Many videos in this dataset contain identical primary objects, respectively. However, there are a few 'ambiguous' videos, in which it is not obvious to pick primary objects. This is because these ambiguous videos contain many kinds of objects whose appearance frequencies are almost the same. The dataset provides ground truth bounding boxes for a selected set of frames that enclose the most distinct objects. Since each video is composed of several shots, we sample 10 frames from each shot and apply the CorLoc metric to the frames whose ground truths are available. Also, as done in [23], we evaluate the discovery performance only for the training videos in the dataset.

In Table 2, we compare the proposed POD algorithm with the Papazoglou and Ferrari's algorithm [22] and the Prest *et al.*'s algorithm [23]. The results of [22, 23] are from the respective papers. The proposed algorithm yields relatively low CorLoc scores on the "cat," "horse," and "train"

Figure 5. Performance comparison of the proposed algorithm with [7,22,29,35]. Results of each algorithm are depicted by bounding boxes of a different color. The small boxes in the left top corner are the ground-truth boxes. A box with a red cross depicts a noisy frame.

classes, which contain relatively many ambiguous videos. On the other hand, most videos in the "boat," "car," "dog," and "motorbike" classes contain clear primary objects, thus POD provides good performances on these classes. On average, as compared with the state-of-the-art algorithm in [22], POD improves the accuracy by 10.1%. Due to the page limitation, we provide primary object discovery results in the supplemental materials.

5. Conclusions

We proposed the POD algorithm for a single video. We first generated object proposals for each frame. Then, we divided each proposal into foreground and background regions, and extracted superpixel-based feature from each region. By combining the foreground and background features, we constructed the object recurrence, background, and primary object models, and iteratively updated each model using the information in the other models. Also, we detected noisy frames during the iteration. Finally, we selected candidate proposals using the primary object model, and localized a primary object by merging the candidate proposals selectively. Experimental results confirmed that the proposed POD algorithm effectively discovers primary objects in the challenging and extensive dataset.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2015R1A2A1A10055037).

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(11):2274–2282, Nov. 2012. 3

[2] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(11):2189–2202, Nov. 2012. 3

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *JMLR*, 3:993–1022, 2003. 2

[4] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, pages 282–295. 2010. 1, 2

[5] K.-Y. Chang, T.-L. Liu, and S.-H. Lai. From co-saliency to co-segmentation: An efficient and fully unsupervised energy minimization model. In *CVPR*, pages 2129–2136, 2011. 2

[6] X. Chen, A. Shrivastava, and A. Gupta. Enriching visual knowledge bases via object discovery and segmentation. In *CVPR*, pages 2035–2042, 2014. 1, 2

[7] M. Cho, S. Kwak, C. Schmid, and J. Ponce. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In *CVPR*, pages 1201–1210, 2015. 1, 2, 6, 7, 8

[8] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 2006. 4

[9] A. Faktor and M. Irani. Video segmentation by non-local consensus voting. In *BMVC*, 2014. 1, 2

[10] D. Giordano, F. Murabito, S. Palazzo, and C. Spampinato. Superpixel-based video object segmentation using perceptual organization and location prior. In *CVPR*, pages 4814–4822, 2015. 1, 2

[11] D. S. Hochbaum and V. Singh. An efficient algorithm for co-segmentation. In *ICCV*, pages 269–276, 2009. 1, 2

[12] A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, pages 1943–1950, 2010. 1, 2

[13] A. Joulin, K. Tang, and L. Fei-Fei. Efficient image and video co-localization with Frank-Wolfe algorithm. In *ECCV*, pages 253–268. 2014. 1, 2, 6

[14] G. Kim, E. P. Xing, L. Fei-Fei, and T. Kanade. Distributed cosegmentation via submodular optimization on anisotropic diffusion. In *ICCV*, pages 169–176, 2011. 2

[15] C. Lee, W.-D. Jang, J.-Y. Sim, and C.-S. Kim. Multiple random walkers and their application to image cosegmentation. In *CVPR*, pages 3837–3845, 2015. 1, 2, 6, 7

[16] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *ICCV*, pages 1995–2002, 2011. 1, 2

[17] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, pages 2192–2199, 2013. 1, 2

[18] T. Ma and L. J. Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *CVPR*, pages 670–677, 2012. 1, 2

[19] L. Mukherjee, V. Singh, and C. R. Dyer. Half-integrality based algorithms for cosegmentation of images. In *CVPR*, pages 2028–2035, 2009. 1, 2

[20] P. Ochs and T. Brox. Object segmentation in video: A hierarchical variational approach for turning point trajectories into dense regions. In *ICCV*, pages 1583–1590, 2011. 2

[21] P. Ochs and T. Brox. Higher order motion models and spectral clustering. In *CVPR*, pages 614–621, 2012. 1, 2

[22] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, pages 1777–1784, 2013. 1, 2, 6, 7, 8

[23] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, pages 3282–3289, 2012. 1, 2, 6, 7

[24] C. Rother, V. Kolmogorov, and A. Blake. GrabCut: Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graphics*, 23(3):309–314, 2004. 3, 5

[25] 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. 1, 2

[26] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu. Unsupervised joint object discovery and segmentation in internet images. In *CVPR*, pages 1939–1946, 2013. 1, 2

[27] J. C. Rubio, J. Serrat, A. López, and N. Paragios. Unsupervised co-segmentation through region matching. In *CVPR*, pages 749–756, 2012. 1, 2

[28] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *ICCV*, pages 1154–1160, 1998. 1, 2

[29] K. Tang, A. Joulin, L.-J. Li, and L. Fei-Fei. Co-localization in real-world images. In *CVPR*, pages 1464–1471, 2014. 1, 2, 6, 7, 8

[30] B. Taylor, V. Karasev, and S. Soatto. Causal video object segmentation from persistence of occlusions. In *CVPR*, pages 4268–4276, 2015. 1, 2

[31] S. Vicente, C. Rother, and V. Kolmogorov. Object cosegmentation. In *CVPR*, pages 2217–2224, 2011. 1, 2

[32] F. Wang, Q. Huang, and L. J. Guibas. Image co-segmentation via consistent functional maps. In *ICCV*, pages 849–856, 2013. 2

[33] L. Wang, G. Hua, R. Sukthankar, J. Xue, and N. Zheng. Video object discovery and co-segmentation with extremely weak supervision. In *ECCV*, pages 640–655. 2014. 1, 2

[34] W. Wang, J. Shen, and F. Porikli. Saliency-aware geodesic video object segmentation. In *CVPR*, pages 3395–3402, 2015. 1, 2

[35] D. Zhang, O. Javed, and M. Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *CVPR*, pages 628–635, 2013. 1, 2, 6, 7, 8

[36] G. Zhao, J. Yuan, and G. Hua. Topical video object discovery from key frames by modeling word co-occurrence prior. In *CVPR*, pages 1602–1609, 2013. 1, 2