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] 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.

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 foreground 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 \( \mathcal{V} = \)}
{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. For each frame, we obtain 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 \( p_{t,i} \) and the background feature \( q_{t,i} \). 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, \( p_{t,i} \) and \( q_{t,i} \), from the foreground and background regions, respectively. For the feature representation, we adopt the bag-of-visual-words approach. Given a video sequence, we oversegment 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 \( p_{t,i} \) is obtained by recording the histogram of the codewords for the foreground region, and the background feature \( q_{t,i} \) is obtained in a similar manner. Both \( p_{t,i} \) and \( q_{t,i} \) 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 \( R_t^{(\theta)}(\Gamma_t) \) at frame \( t \) at iteration \( \theta \) as

\[
R_t^{(\theta)}(\Gamma_t) = \sum_{i=1}^{m} \gamma_{t,i} p_{t,i}.
\]

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

To obtain \( \Gamma_t \), 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 \( \mathcal{P}^{(\theta)} = (\mathcal{P}_1^{(\theta)}, \ldots, \mathcal{P}_T^{(\theta)}) \) denote the set of the primary object models at iteration \( \theta \). At the start of iterations, the primary object model \( \mathcal{P}_t^{(0)} \) at frame \( t \) is initialized to the average of the foreground features, given by

\[
\mathcal{P}_t^{(0)} = \frac{1}{m} \sum_{i=1}^{m} p_{t,i}.
\]
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 $\theta$ in (b)~(g), from top to bottom, the object recurrence model $R_\theta^t(\Gamma_t^\theta)$, the background model $B_\theta^t$, and the primary object model $P_\theta^t$ are shown. To visualize $R_\theta^t(\Gamma_t^\theta)$ and $P_\theta^t$, the foreground regions of proposals are linearly superposed using weights $\gamma_{t,i}$ and $\omega_{t,i}$, respectively. Similarly, for $B_\theta^t$, the background regions are superposed using weights $\lambda_{t,i}$.

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 $P^{(\theta-1)}$ as

$$P^{(\theta-1)} = \sum_{t=1}^{T} c_{t}^{(\theta-1)} P_{t}^{(\theta-1)} \sum_{t=1}^{T} c_{t}^{(\theta-1)} \quad (3)$$

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

$$D(R_t^\theta(\Gamma_t^\theta)||P^{(\theta-1)}) \quad (4)$$

Note that the Kullback-Leibler divergence or relative entropy

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

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

$$\Gamma_t^\theta = \arg \min_{\Gamma_t^\theta} D(R_t^\theta(\Gamma_t^\theta)||P^{(\theta-1)}) \quad (6)$$

subject to

$$0 \leq \gamma_{t,i} \leq 1, \sum_i \gamma_{t,i} = 1. \quad (7)$$

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 $\Gamma_t^\theta$, we obtain the object recurrence model $R_t^\theta(\Gamma_t^\theta)$ at frame $t$. Note that the optimal $\Gamma_t^\theta$ makes the object recurrence model $R_t^\theta(\Gamma_t^\theta)$ as close to the global primary object model $P^{(\theta-1)}$ as possible.

Background Model: Next, we define the background model $B_t^\theta$ at frame $t$ at iteration $\theta$ using the background features of the proposals as

$$B_t^\theta = \sum_{i=1}^{m} \frac{\lambda_{t,i} q_{t,i}}{\sum_{i=1}^{m} \lambda_{t,i}} \quad (8)$$
where $\lambda_{t,i}$ is the weight for the background feature $q_{t,i}$ of the $i$th proposal. The background model should be distinguishable from the object recurrence model. Therefore, we assign a higher weight $\lambda_{t,i}$, when the feature $q_{t,i}$ is more dissimilar from $R_t^{(\theta)}(\Gamma_t^*)$. In other words, we set the weight $\lambda_{t,i}$ as

$$\lambda_{t,i} = d_\chi(q_{t,i}, R_t^{(\theta)}(\Gamma_t^*))$$

(9)

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

**Primary Object Model:** Even though the object recurrence model $R_t^{(\theta)}(\Gamma_t^*)$ 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 $p_{t,i}$ of the $i$th proposal as

$$\omega_{t,i} = \frac{d_\chi(p_{t,i}, B_t^{(\theta)})}{d_\chi(p_{t,i}, R_t^{(\theta)}(\Gamma_t^*))}.$$  

(10)

A high $\omega_{t,i}$ indicates that the foreground feature $p_{t,i}$ 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 $P_t^{(\theta)}$ at frame $t$ at the current iteration $\theta$, by superposing the foreground features of the selected proposals,

$$P_t^{(\theta)} = \frac{\sum_{i \in T_t} \omega_{t,i} p_{t,i}}{\sum_{i \in T_t} \omega_{t,i}}$$  

(11)

where $T_t$ is the index set of the top-5 proposals at frame $t$. Note that these primary object models $P^{(\theta)} = (P_t^{(\theta)}, \ldots, P_T^{(\theta)})$ 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 $P^{(\theta)}$ in (3) and each model $P_t^{(\theta)}$ in (11). We declare frame $t$ as noisy if $d_\chi(P_t^{(\theta)}, P^{(\theta-1)})$ is larger than a threshold 0.7. This information is recorded in the indicator vector $e^{(\theta)} = (c_1^{(\theta)}, \ldots, c_T^{(\theta)})$. Here, $c_t^{(\theta)} = 0$ 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 $d(P_t^{(\theta)}, P^{(\theta-1)})$ 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.

**3.2. Discovering a Primary Object**

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

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

$$\delta = \arg\min_{i \in T_t} d_\chi(p_{t,i}, \bar{P}).$$

(12)

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

Let $F_\delta$ and $F_i$ 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 $F_i \cup F_\delta$ in Figure 4(c) and the difference region $F_i \setminus F_\delta$ in Figure 4(d). Then, we extract the features $p_{0,\delta}$ and $p_{0,i}$ from the union region and the difference region, respectively. Using these features, we evaluate the score function as

$$\Psi(o_{t,i}, o_{t,\delta}) = \frac{d_\chi(p_{t,i}, \bar{P})}{d_\chi(p_{t,\delta}, \bar{P})} \left(1 - d_\chi(p_{t,\delta}, p_{t,i})\right).$$

(13)

Suppose that the foreground region $F_i$ 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 $p_{0,i}$ of the union region should be similar to the global primary object model $\bar{P}$, as the feature $p_{0,\delta}$ of the main proposal is. In other words, the ratio $d_\chi(p_{t,i}, \bar{P})/d_\chi(p_{t,\delta}, \bar{P})$ should be close to 1. Moreover, the difference region should have a similar feature to the main proposal, and $d_\chi(p_{t,\delta}, p_{t,i})$ should be
small. Note that the chi-square distance ranges from 0 to 1. Thus, if

small. Note that the chi-square distance ranges from 0 to 1.

Thus, if

small. Note that the chi-square distance ranges from 0 to 1.

Thus, if

small. Note that the chi-square distance ranges from 0 to 1.

Thus, if

small. Note that the chi-square distance ranges from 0 to 1.

Thus, if

small. Note that the chi-square distance ranges from 0 to 1.

Thus, if

small. Note that the chi-square distance ranges from 0 to 1.

Thus, if

small. Note that the chi-square distance ranges from 0 to 1.

Thus, if

small. Note that the chi-square distance ranges from 0 to 1.

Thus, if

small. Note that the chi-square distance ranges from 0 to 1.

Thus, if

small. Note that the chi-square distance ranges from 0 to 1.

Thus, if

small. Note that the chi-square distance ranges from 0 to 1.

Thus, if

small. Note that the chi-square distance ranges from 0 to 1.

Thus, if

small. Note that the chi-square distance ranges from 0 to 1.

Thus, if

small. Note that the chi-square distance ranges from 0 to 1.

Thus, if

small. Note that the chi-square distance ranges from 0 to 1.

Thus, if

small. Note that the chi-square distance ranges from 0 to 1.

Thus, if

small. Note that the chi-square distance ranges from 0 to 1.

Thus, if

small. Note that the chi-square distance ranges from 0 to 1.

Thus, if

small. Note that the chi-square distance ranges from 0 to 1.

Thus, if

small. Note that the chi-square distance ranges from 0 to 1.

Thus, if

small. Note that the chi-square distance ranges from 0 to 1.

Thus, if

small. Note that the chi-square distance ranges from 0 to 1.

Thus, if

small. Note that the chi-square distance ranges from 0 to 1.

Thus, if

small. Note that the chi-square distance ranges from 0 to 1.

Thus, if

small. Note that the chi-square distance ranges from 0 to 1.

Thus, if

small. Note that the chi-square distance ranges from 0 to 1.

Thus, if

small. Note that the chi-square distance ranges from 0 to 1.

Thus, if

small. Note that the chi-square distance ranges from 0 to 1.

Thus, if

small. Note that the chi-square distance ranges from 0 to 1.

Thus, if
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 “Helicopter” 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 primary objects that enclose the most distinct objects. Since each video contains 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” objects, 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.

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

<table>
<thead>
<tr>
<th>Category</th>
<th>Video (No. of frames)</th>
<th>Cosegmentation</th>
<th>Object discovery</th>
<th>Video object segmentation</th>
<th>Proposed POD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>Baby (528)</td>
<td>71.00</td>
<td>14.84</td>
<td>25.24</td>
<td><strong>82.96</strong></td>
</tr>
<tr>
<td></td>
<td>Helicopter (200)</td>
<td>26.03</td>
<td>12.50</td>
<td>67.00</td>
<td>74.19</td>
</tr>
<tr>
<td></td>
<td>Rabbit (459)</td>
<td><strong>93.31</strong></td>
<td>64.54</td>
<td>73.13</td>
<td>92.73</td>
</tr>
<tr>
<td></td>
<td>Cat (299)</td>
<td><strong>93.00</strong></td>
<td>31.44</td>
<td>49.83</td>
<td>92.98</td>
</tr>
<tr>
<td></td>
<td>Babypanda (361)</td>
<td>73.09</td>
<td>52.15</td>
<td>53.58</td>
<td><strong>94.18</strong></td>
</tr>
<tr>
<td>Static</td>
<td>Yellow Bear (147)</td>
<td>3.47</td>
<td>27.59</td>
<td>40.00</td>
<td><strong>99.32</strong></td>
</tr>
<tr>
<td></td>
<td>Raccoon (586)</td>
<td>49.25</td>
<td>38.83</td>
<td>45.02</td>
<td>72.18</td>
</tr>
<tr>
<td></td>
<td>RC Car (287)</td>
<td>78.78</td>
<td>28.92</td>
<td>70.04</td>
<td><strong>88.50</strong></td>
</tr>
<tr>
<td></td>
<td>Polarbear (247)</td>
<td>95.89</td>
<td>55.42</td>
<td>44.58</td>
<td>84.58</td>
</tr>
<tr>
<td></td>
<td>Fox (393)</td>
<td>87.00</td>
<td>58.93</td>
<td>16.07</td>
<td><strong>88.30</strong></td>
</tr>
<tr>
<td>Multi-objects</td>
<td>Monkey (131)</td>
<td>58.00</td>
<td>30.53</td>
<td>36.64</td>
<td><strong>80.92</strong></td>
</tr>
<tr>
<td></td>
<td>Panda (271)</td>
<td>53.72</td>
<td>42.74</td>
<td>41.91</td>
<td>95.44</td>
</tr>
<tr>
<td></td>
<td>Car (322)</td>
<td>66.08</td>
<td>40.52</td>
<td>66.34</td>
<td><strong>66.99</strong></td>
</tr>
<tr>
<td></td>
<td>Baby2 (349)</td>
<td>71.00</td>
<td>10.32</td>
<td>34.10</td>
<td><strong>87.97</strong></td>
</tr>
<tr>
<td></td>
<td>Peacock (217)</td>
<td>64.14</td>
<td>24.17</td>
<td>45.02</td>
<td>74.41</td>
</tr>
<tr>
<td></td>
<td>Fox (355)</td>
<td>20.56</td>
<td>17.39</td>
<td>10.44</td>
<td><strong>61.13</strong></td>
</tr>
<tr>
<td>Animation</td>
<td>PooH (387)</td>
<td>42.00</td>
<td>14.21</td>
<td>3.88</td>
<td><strong>93.80</strong></td>
</tr>
<tr>
<td></td>
<td>Yellow Larva (280)</td>
<td>4.00</td>
<td>33.57</td>
<td>6.43</td>
<td><strong>87.50</strong></td>
</tr>
<tr>
<td></td>
<td>Frozen (434)</td>
<td>9.00</td>
<td>10.00</td>
<td>9.07</td>
<td><strong>64.88</strong></td>
</tr>
<tr>
<td></td>
<td>Doosly (450)</td>
<td>54.00</td>
<td>11.33</td>
<td>16.22</td>
<td><strong>83.56</strong></td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>54.28</td>
<td>31.00</td>
<td>35.90</td>
<td>43.96</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>83.44</strong></td>
</tr>
</tbody>
</table>
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


