CATS: Co-saliency Activated Tracklet Selection for Video Co-localization

Conference Paper · October 2016

DOI: 10.1007/978-3-319-46478-7_12

3 authors:

Koteswar Rao Jerripothula
Graphic Era University
19 PUBLICATIONS **75** CITATIONS
SEE PROFILE

Jianfei Cai
Nanyang Technological University
286 PUBLICATIONS **2,968** CITATIONS
SEE PROFILE

Junsong Yuan
Nanyang Technological University
246 PUBLICATIONS **6,875** CITATIONS
SEE PROFILE

CATS: Co-saliency Activated Tracklet Selection for Video Co-localization

Koteswar Rao Jerripothula[1,2], Jianfei Cai[2], and Junsong Yuan[3]

[1]Interdisciplinary Graduate School,
[2]School of Computer Science and Engineering,
[3]School of Electrical and Electronic Engineering,
Nanyang Technological University, Singapore
{koteswar001,asjfcai,jsyuan}@ntu.edu.sg

Abstract. Video co-localization is the task of jointly localizing common objects across videos. Due to the appearance variations both across the videos and within the video, it is a challenging problem to identify and track them without any supervision. In contrast to previous joint frameworks that use bounding box proposals to attack the problem, we propose to leverage *co-saliency activated tracklets* to address the challenge. To identify the common visual object, we first explore inter-video commonness, intra-video commonness, and motion saliency to generate the co-saliency maps. Object proposals of high objectness and co-saliency scores are tracked across short video intervals to build tracklets. The best tube for a video is obtained through tracklet selection from these intervals based on confidence and smoothness between the adjacent tracklets, with the help of dynamic programming. Experimental results on the benchmark YouTube Object dataset show that the proposed method outperforms state-of-the-art methods.

Keywords: tracklet, co-localization, co-saliency, co-detection, video, cats.

1 Introduction

Localizing the common object in a video is an important task in computer vision since it facilitates many other vision tasks such as object recognition and action recognition. Recent research interests have been shifted from single-video object localization to video co-localization [13, 14], which aims at jointly localizing common objects across videos by exploiting shared attributes among videos as weak supervision.

Video co-localization is a challenging problem due to the following reasons. First, for a large diverse video dataset, it is non-trivial to discover the related videos that contain semantically similar objects. Second, even for videos from the same semantic class, their common objects may exhibit large inter-video variations (see Fig. 1(a)). Third, even within one video, objects could also have large variations due to viewpoint/pose changes (see Fig. 1(b)).

A few video co-localization works [13, 24] have been proposed in literature. In particular, [13] proposed to co-select bounding box proposals, and [24] proposed

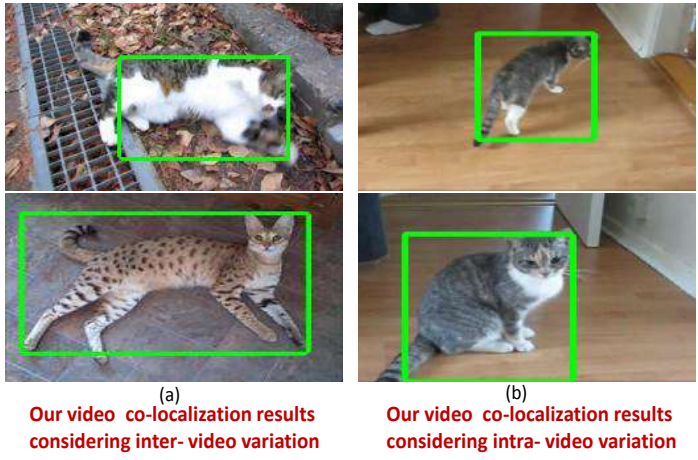

Fig. 1. Variations of cats (a) across the videos as well as (b) within the video make the co-localization problem very challenging.

to co-select tubes across the videos. Both methods try to localize common objects in multiple videos simultaneously. Surprisingly, such joint processing methods did not outperform the individual video processing based framework [23]. One reason could be the inability of both methods to handle large variations of objects across the videos in the same class. Such an observation that co-processing might not be better than individual processing has also been reported in some relevant studies [27, 26, 13]. This motivates us to propose a framework to divide video co-location into two steps: exploiting inter-video relationship to find the common object prior and then locating the common object separately in each individual video, in other words, we propose a guided single video-based framework. Similar to this idea, recently [14] developed a two-step framework for video co-localization, where they iteratively discover common objects across neighboring videos and then incorporate the prior into individual video localization. However, [14] relies on bounding box proposals independently extracted at every sampled frame, which itself could be quite noisy.

Instead of relying on large number of bounding box proposals, in this paper we propose to leverage *co-saliency activated tracklets* for video co-localization. In particular, we first explore inter-video commonness, intra-video commonness, and motion saliency to generate the co-saliency maps and then fuse them to extract object prior masks for uniformly sampled key frames. We then make use the object prior to select only a small set of proposals at each key frame and use them to activate the tracklets to be generated across subsequent frames. Finally, we separately generate the best tube for each video by selecting optimal tracklets based on confidence and consistency between adjacent tracklets using dynamic programming. Experimental results on the benchmark YouTube Object dataset show that our proposed method outperforms state-of-the-art methods.

We would like to point out that our work is also motivated by benefits of co-saliency and tracklets. Co-saliency research [29][28][17] has recently demonstrated significant contribution in object discovery problems. On the other hand, tracklets developed through trackers [18][19][2][20][21][3] are quite spatio-temporally consistent and reliable already for short video intervals. In addition, tracklet processing is much more efficient than bounding box based processing [13] [14].

The main contributions of this paper are twofold: 1) exploring inter-video, intra-video and motion information for tracklet activations; 2) leveraging tracklets for video co-localization.

2 Related Work

Our work is closely related to video co-localization, co-saliency topics, and, therefore, we briefly discuss them in this section.

2.1 Video Co-localization

Video Co-localization is a task of jointly localizing the shared object in a set of videos. The recent work of [13] and [24] proposed joint framework to locate common objects across video. In [13], it used Quadratic Programming framework to co-select bounding box proposals in all the frames in all the videos together. While in [24], it formed candidate tubes and co-selected tubes across the videos to locate the shared object. Handling inter-video, intra-video variations and temporal consistency simultaneously often become difficult task for such joint frameworks. This is especially so when extremes such as bounding box in a frame or candidate tube for entire video is chosen as processing unit. Contrary to these, short interval tracklets can also be considered as an alternative. Also considering an individual processing framework can simplify things provided a guiding object prior is available such as co-saliency. Recently, [14] proposed an approach of developing foreground confidence for bounding boxes and selecting bounding boxes while maintaining temporal consistency. Presence of noisy bounding box proposals mandates taking an iterative approach in [14]. This can be avoided if we have a guiding object prior like co-saliency for filtering purposes to have a one shot method. Also considering tracklets over bounding boxes can significantly reduce computational complexity. All these methods [13][14][24] assumed the object is present in all the frames in all the videos, but [28] overcame such an assumption through providing few labels of relevant frames and irrelevant frames to effectively guide object discovery. Similarly, in this paper, we attempt to guide the object discovery process, but through co-saliency instead of human intervention.

2.2 Co-Saliency

Co-saliency generally refers to the common saliency that exists in a set of images containing similar objects. This term was first introduced by [8], in the

sense of what is unique in a set of very similar images (e.g. two back-to-back snapshots), and this concept was later linked to extracting common saliency, which has many practical applications [6, 15]. In [5], co-saliency object priors have been effectively used in the co-segmentation problem. A cluster based co-saliency method using various cues was proposed in [7], which learns the global correspondence and obtains cluster saliency quite well. However, [7] was mainly designed for images of the same (or very similar) object instances captured at different viewpoints or time. Therefore, it cannot struggles to handle image sets with huge intra-class variation. In [11], it fused saliency maps from different images *via* warping technique and it claims to handle the intra-class variation well, which was then extended to [10] for the large scale application. In [29], it introduced deep intra-group semantic information and wide cross-group heterogeneousness information for co-saliency detection. In this way, they can capture the concept-level properties of the co-salient objects and suppress the common backgrounds in the image group. Co-saliency maps are able to provide good object priors and, therefore, we rely on it for the activation of tracklets.

3 Proposed Method

Our framework consists of three major steps: co-saliency based object prior generation, tracklet activation and generation, and tube generation, as shown in Fig. 2. First of all, each video is uniformly cut into short-interval video trunks and in each video trunk, we generate some tracklets, each of which is a sequence of bounding boxes across consecutive frames, hoping to locate the common object with high recall. Since each tracklet needs an initial bounding box at its starting frame (we call such starting frame an activator), the first step of our framework is to generate a co-saliency map for each activator so as to provide some object prior information. The second step is to make use of the object prior mask to generate good initial bounding boxes and the corresponding tracklets, from which we generate a set of tracklets between every two adjacent activators. Finally, the third step of our framework is to select one tracklet per set to form a tube which localizes the object. We name our framework *co-saliency activated tracklet selection* (CATS).

3.1 Co-saliency Based Object Prior Generation

To generate good object prior, our basic idea is to combine the following three type of co-saliency. 1) Inter-video co-saliency: since one video of a common object often contains similar background, it is needed to introduce other videos of similar objects that are likely to have different backgrounds. Thus, we exploit the activators from different videos of similar objects to obtain inter video co-saliency. 2) Intra video co-saliency: Sometimes the activators from the same video could also contain diverse backgrounds, from which we could highlight intra-video co-saliency. 3) Total motion saliency: Since motion clues are always critical for video analysis, we want to use motion to identify co-saliency among

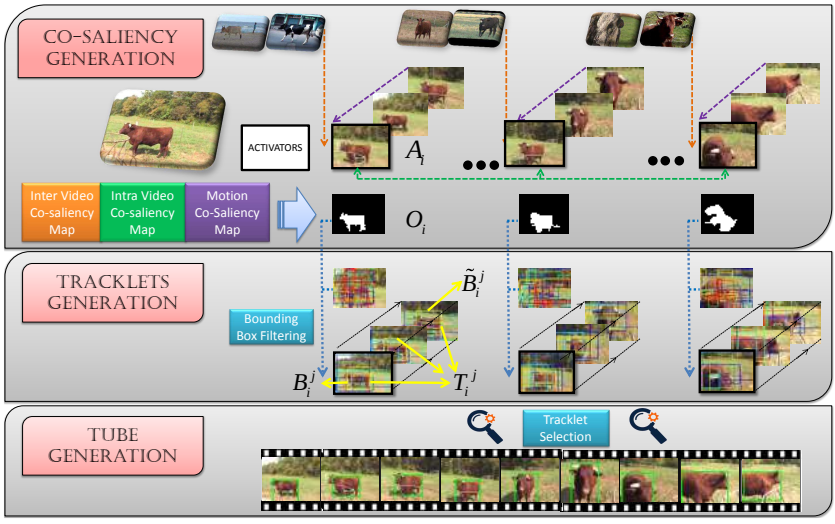

Fig. 2. Overview of the proposed *co-saliency activated tracklet selection* (CATS) for video co-localization, which consists of three main components: co-saliency generation, tracklet generation and tube generation. NOTE: 3 different co-saliency processes are represented in 3 different colors: (1) inter-video (orange), (2) intra-video (green), and (3) motion (violet). Bounding boxes of same color across a video trunk denote a tracklet.

consecutive frames. Once the three co-saliency maps are obtained, we fuse them by averaging followed by segmentation to obtain a co-saliency based object mask for each activator for the subsequent tracklet generation.

Inter video co-saliency: Let $\mathcal{A} = \{A_1, A_2, ..., A_n\}$ be a set of n activators (uniformly sampled) in a video $\mathcal{V}$ such that $\mathcal{A} \subseteq \mathcal{V}$, where $\mathcal{V}$ is the set of all the frames in the video. Let $\mathbb{V}$ be the set of similar videos (containing a similar semantic object) such that $\mathcal{V} \in \mathbb{V}$. For each activator, say A_i, we search for its matched activators from other videos in $\mathbb{V}$ to create a externally matched activators set $\mathcal{N}_i^{ext} = \{\mathbb{A} | \delta(A_i, \mathbb{A}) < \epsilon, \mathbb{A} \in \mathbb{V} \backslash \mathcal{V}\}$, where $\mathbb{A}$ denotes externally matched activator and δ denotes distance function. Particularly, we extract the GIST descriptor [22] from each activator weighted by its initial saliency map [12]. The distance $\delta(A_i, \mathbb{A})$ between a pair of activators is measured as the l_2 distance between their weighted GIST features. Such distance computation is essentially to find the activators that contain similar saliency regions. For an activator A_i, once its externally matched activators set $\mathcal{N}_i^{ext}$ is obtained, we compute the inter-video co-saliency M_i^{ext} as

$$M_i^{ext} = \frac{\mathcal{S}(A_i) + \sum\limits^{\mathbb{A} \in \mathcal{N}_i^{ext}} \mathcal{W}_{\mathbb{A}}^{A_i}\big(\mathcal{S}(\mathbb{A})\big)}{|\mathcal{N}_i^{ext}| + 1} \tag{1}$$

where $\mathcal{S}(\cdot)$ denotes the initial saliency map filter, $\mathcal{W}_{\mathbb{A}}^{A_i}(\cdot)$ denotes warping function from $\mathbb{A}$ to A_i, and $|.|$ denotes cardinality. We use the masked dense SIFT correspondence (SIFT flow) [16, 26] to find pixel correspondences for the warping. Eq. (1) essentially computes the joint saliency of the matched object points in different activators by such average of own saliency and warped saliency maps.

Intra video co-saliency: We obtain intra-video co-saliency in a similar way as that for inter-video co-saliency. Particularly, we first group the activators in one video into different clusters using k-means based on weighted GIST descriptor as discussed before. Then, for an activator, other activators in its cluster are considered as its matches. Therefore, internally matched activators set $\mathcal{N}_i^{int} = \{A_j | A_j \in C_k \backslash A_i, A_i \in C_k\}$ is basically all other activators in cluster C_k to which A_i belongs after the clustering. The intra-video co-saliency M_i^{int} for activator A_i is also computed as the average of its own saliency and the warped saliency maps of its matches, i.e.

$$M_i^{int} = \frac{\mathcal{S}(A_i) + \sum\limits^{A_j \in \mathcal{N}_i^{int}} \mathcal{W}_{A_j}^{A_i}(\mathcal{S}(A_j))}{|\mathcal{N}_i^{int}| + 1} \tag{2}$$

where definitions of $\mathcal{S}$ and $\mathcal{W}$ remain same as defined previously. Here, for applying SIFT flow to find pixel correspondences for warping, we use not only SIFT feature but also color features (RGB, HSV, and Lab) since the common object in one video is likely to be of similar color.

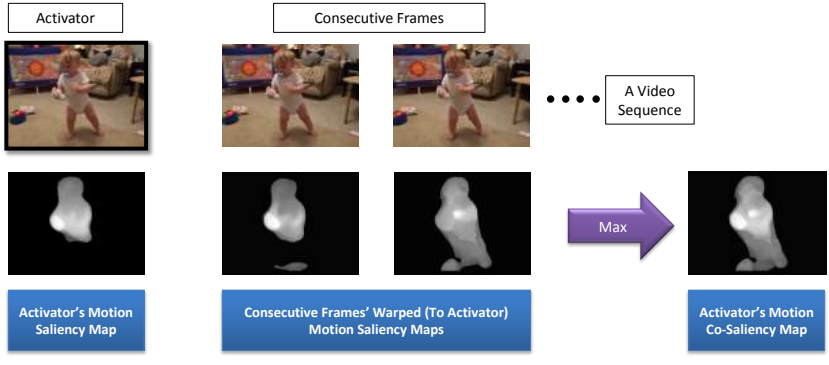

Fig. 3. Motion co-saliency: Considering non-rigid object motion, max pooling motion saliency of different parts at different frames help develop a proper object prior.

Motion Co-saliency: For an activator, many subsequent frames are generally similar to it, typically with some variations due to object movements. We adopt the $\omega - flow$ method in [9] to extract the motion saliency map for each

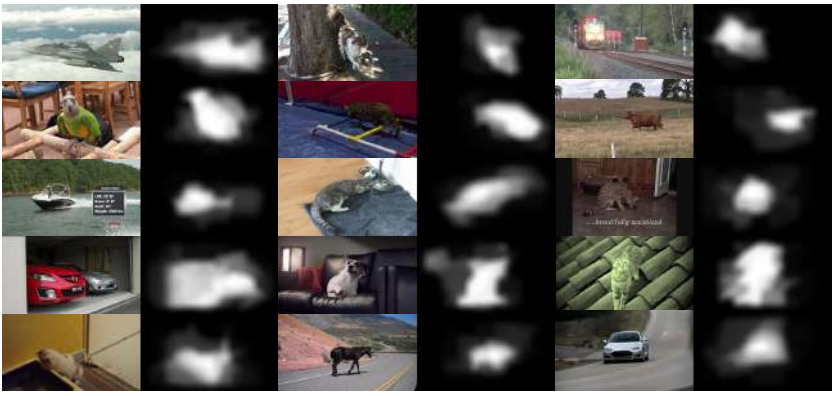

Fig. 4. Final fused co-saliency maps for some activator samples in YouTube-Object dataset

frame in a video trunk. Considering that for deformable objects, parts of the object could move while other parts might remain still (see Fig. 3 for example), we propose to use max pooling to collect motion saliency from an activator and its consecutive frames after warping, which we call *motion co-saliency* M_i^{mot}, defined as

$$M_i^{mot} = \max\left(\mathcal{M}(A_i), \max_{I_j \in \mathcal{N}_i^{mot}} \left(\mathcal{W}_{I_j}^{A_i}\left(\mathcal{M}(I_j)\right)\right)\right) \tag{3}$$

for activator A_i, where $\mathcal{M}$ denotes the motion saliency filter, set $\mathcal{N}_i^{mot} = \{I_j | I_j \in \mathcal{V}[A_i, A_{i+1}]\}$ denotes consecutive frames of activator A_i, i.e. between A_i and A_{i+1}, and $\max(\cdot)$ denotes pixel-level maximum function.

Generating object prior: We simply fuse the three co-saliency maps, namely inter video co-saliency map (M_i^{ext}), intra video co-saliency map (M_i^{int}) and motion co-saliency map (M_i^{mot}), through averaging so that possible saliency defects which may exist in the individual maps can get subdued in the fused one. Once the final fused co-saliency map is available (see Fig. 4 for examples), we apply the GrabCut [25] to obtain a binary segmentation mask, denoted as object prior O_i, for activator A_i.

3.2 Tracklet Activation and Generation

Bounding box filtering: We need an initial bounding box at the activator to activate a tracklet which then ends at next activator. Following state-of-the-art methods [23, 14], we also use bottom-up object proposal techniques, particularly [1], to generate initial bounding boxes. However, to ensure a high object detection rate, the existing general object proposal technique typically requires to generate at least hundreds of proposals, which makes the subsequent tracklet generation and tube generation infeasible. Thus, we propose to make use of our

generated co-saliency based object prior to greatly trim down a large number of object proposals.

Particularly, we rank each object proposal by its objectness score [1] and its overlap with the tight bounding box of the co-saliency based object prior. Let B_i^o denote the tight bounding box of the largest component in the object prior O_i and B_i^j be an object proposal in activator A_i. We calculate an object confidence score X for proposal B_i^j as

$$X(B_i^j) = X_o(B_i^j) + J(B_i^j, B_i^o) \tag{4}$$

where $X_o(B_i^j)$ is the objectness score (between 0 and 1) directly obtained from [1] and $J(\cdot)$ is Jaccard similarity function (also called IoU, intersection over union). We then select the top-m proposals with highest confidence scores.

Tracklet confidence scores: Once m candidate bounding box proposals are selected at the activator, tracklets are obtained using the existing tracker [3] starting from these proposals at the activator and ending at the next activator, which we call *co-saliency activated tracklets*. Let T_i^j denote a tracklet activated at A_i by B_i^j and ending at A_{i+1} with bounding box $\tilde{B}_i^j$. To facilitate the subsequent tube generation via tracklet selections, for tracklet T_i^j, we define two confidence scores based on its IoU values with the object prior bounding boxes at A_i and A_{i+1}, respectively:

$$X_f(T_i^j) = J\big(B_i^j, B_i^o\big), \tag{5}$$

$$X_l(T_i^j) = J\big(\tilde{B}_i^j, B_{i+1}^o\big) \tag{6}$$

where X_f and X_l are defined as first and the last confidence scores of a tracklet based on our object priors at its two ends, respectively. Since we don't have objectness score (X_o) for the last bounding box produced by tracking, we omit the use of objectness score here altogether, even for first bounding box, although available.

3.3 Tube Generation

Given the n sets of tracklets from n activators in a video, we need to select one tracklet from each set to create a spatio-temporal consistent tube which localizes the common object with high confidence. Let $\mathcal{T} = \{T_1, T_2, \ldots, T_n\}$ be a possible tube. Our goal is to find the best tube for every video that minimizes the following criterion, i.e.

$$\min \sum_{i=1}^{n-1} -\log\Big(X_l(T_i)X_f(T_{i+1})\Big) - \lambda \log\Big(J\big(\tilde{B}_i, B_{i+1}\big)\Big) \tag{7}$$

where tracklet T_i starts with B_i and ends with $\tilde{B}_i$, and λ is a trade-off parameter. At any activator (A_{i+1}), both the selected adjacent tracklets (T_i, T_{i+1}) should have high confidence scores. Therefore, the first term in Eq. (7) is to measure how

confidently a pair of adjacent tracklets T_i and T_{i+1} contain the object w.r.t. the object prior B^o_{i+1}. The selected adjacent tracklets (T_i, T_{i+1}) should also overlap well with each other to form a consistent tube. Therefore, the second term in Eq. (7) is to measure the smoothness between the adjacent tracklets via their IoU value. While one term signifies the reliance on co-saliency, another term signifies the reliance on temporal consistency between activated tracklets, to perform what we call as *co-saliency activated tracklet selection* (CATS), resulting in video co-localization. This problem of Eq. (7) can be well solved using dynamic programming.

4 Experimental Results

We evaluate our method on the benchmark YouTube Object Dataset using the evaluation metric of CorLoc, which is defined as the percentage of frames that satisfies the IoU condition: $\frac{area(B_{gt} \cap B_{co})}{area(B_{gt} \cup B_{co})} > 0.5$, where B_{gt} and B_{co} are ground-truth and computed bounding boxes, respectively. YouTube Object Dataset consists of videos downloaded from YouTube and is divided into 10 object classes. Each object class consists of several video shots of the objects belonging to the class. We treat each shot as a video sequence and group all the shots in one class as a weakly supervised scenario for video co-localization.

4.1 Implementation Details

Activators are chosen at the interval of 50 frames. While calculating inter video co-saliency, we wanted to ensure that at least 10 best matched activators should be available, therefore we used K-NN instead of ϵ-NN algorithm. For intra-video co-saliency, we set the number of clusters as $\{n/10\}$ where n is the total number of activators in a video and $\{\cdot\}$ denotes the rounding function. We use [12] to generate saliency maps for individual activators. We choose $m = 10$ at bounding-box filtering step, and sample every 5^{th} frame between activators to generate total motion saliency map for preceding activator to avoid repetitiveness. The parameter λ introduced in Eq. (7), i.e. weight for temporal consistency, is set to 2, same as [14]. For the off shelf techniques we adopt including tracklets [3], motion saliency [9] and GrabCut [25], we use their default settings.

4.2 Co-localization Performance

Results under weakly supervised scenarios: Table. 1 shows the CorLoc performance on YouTube Object Dataset under weakly supervised scenarios using our full-fledged CATS method (*ext+int+mot*), where *ext*, *int* and *mot* refer to using inter-video co-saliency, intra-video co-saliency and motion co-saliency respectively for obtaining the final co-saliency map. We compare with state-of-the-art methods on video co-localization. It can be seen that we almost double the average performance of the frameworks [24] and [13] that simultaneously locate the common object in multiple videos. This suggests that single video

localization with an incorporated object priors from other videos is better than directly performing co-localization on multiple videos, since inter-video variations could be huge. Moreover, thanks to our proposed co-saliency generation and the adoption of consistent tracklets, we achieve 4.5% improvement for the average performance over the state-of-the-art [14], which uses bounding box proposals at every frame and optimizes over them to obtain consistency. Compared to bounding box proposals, using tracklets significantly reduces the computational complexity as the number of nodes to deal with are drastically reduced. For example, for a video of 1000 frames, [14] would need to deal with 100×50 nodes (according to their settings of 100 selected proposals per key frame and 1 sampled key frame per 20 frames), whereas we only need to deal with only 10×20 nodes (default 10 proposals/activator and 1 sampled activator per 50 frames). Considering that more noisy nodes are eliminated, it results in more reliable results. In addition, [14] is an iterative approach and needs 5 iterations to achieve as good as 55.7% score beginning with nearly 38% score at the first iteration, whereas our method achieves 58.2% score in just one shot.

Table 1. CorLoc results of video co-localization on YouTube Object Datatset under weakly supervised scenarios.

	aeroplane	bird	boat	car	cat	cow	dog	horse	motorbike	train	avg
[24]	51.7	17.5	34.4	34.7	22.3	17.9	13.5	26.7	41.2	25.0	28.5
[13]	25.1	31.2	27.8	38.5	41.2	28.4	33.9	35.6	23.1	25.0	31.0
[14]	56.5	**66.4**	58.0	**76.8**	39.9	**69.3**	50.4	**56.3**	**53.0**	31.0	55.7
ext	62.4	43.3	63.8	50.9	51.9	63.8	61.7	43.4	30.0	45.7	51.7
int+mot	64.7	48.1	60.9	54.5	51.2	64.0	58.9	42.5	27.0	**46.6**	51.8
ext+int+mot	**65.7**	59.6	**66.7**	72.3	**55.6**	64.6	**66.0**	50.4	39.0	42.2	**58.2**

In addition to our full-fledged method, in Table. 1 we also show the results of the variants (*ext*, *int+mot*) that use different combinations of the co-saliency maps. The results of *ext* show how much we can explore other videos to help the localization in the considered video. The results of *int+mot* show how much we can benefit from the single video itself. We can see that the combination of all the three co-saliency maps achieves the best performance.

In Fig. 5, we show the localization results (red) on some of the frames in the dataset along with their ground truths (green). It can be seen that our proposed method is able to effectively localize the dominant objects with various poses and shapes across the videos. In Fig. 6, we demonstrate our localization results on different videos. It can be seen that our method is able to effectively handle various pose variations in the videos of the car, cat and horse, the size variation in cow and motorbike, and the location variation in dog video. At the same time, our method is also able to handle objects that do not move much such as in the video of bird. These results clearly demonstrate the robustness of our method in different scenarios.

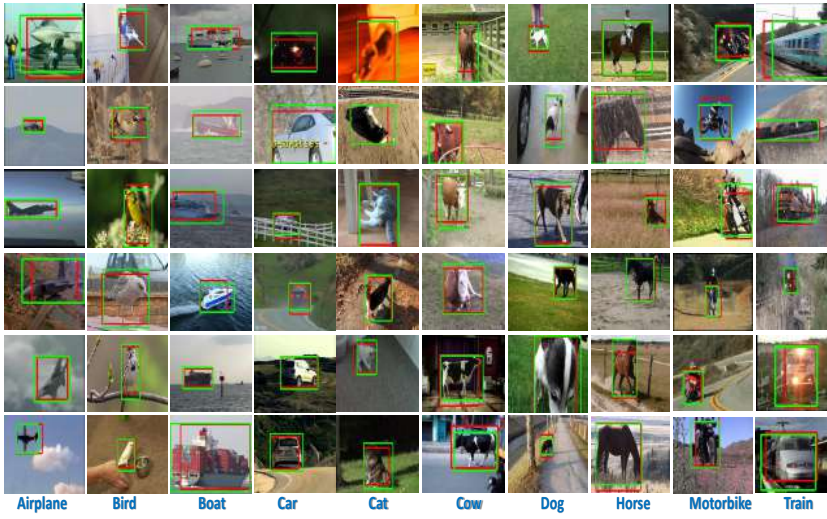

Fig. 5. Sample localization results (red) along with groundtruths (green) on YouTube Objects dataset.

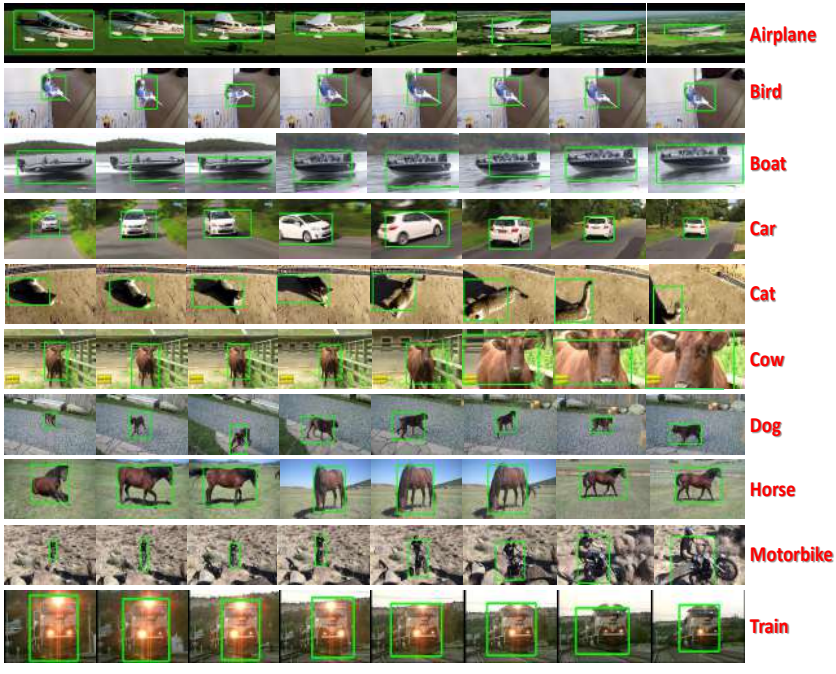

Fig. 6. Our video co-localization results on YouTube Object Datatset. It can be seen that our method can handle variations in size (for airplane, cow and motorbike), position (for dog), pose (for car, cat and horse), and mobility (negligible motion for bird).

Table 2. CorLoc results on YouTube Object Datatset in an unsupervised scenario where we do not use class labels.

	aeroplane	bird	boat	car	cow	cat	dog	horse	motorbike	train	avg
[4]	53.9	19.6	38.2	37.8	32.2	21.8	27.0	34.7	45.4	37.5	34.8
[23]	65.4	**67.3**	38.9	65.2	46.3	40.2	**65.3**	48.4	39.0	25.0	50.1
[14]	55.2	58.7	53.6	**72.3**	33.1	58.3	52.5	**50.8**	**45.0**	19.8	49.9
ext+int+mot	**66.7**	48.1	**62.3**	51.8	**49.6**	**60.6**	58.9	41.9	28.0	**47.4**	**51.5**

Table 3. The recall performance on YouTube Object Dataset using either the existing objectness scores $X_o(B_i^j)$ [1] or the proposed object confidence scores $X(B_i^j)$ in (4) for bounding box selection.

	Top-1	Top-3	Top-5	Top-10	Top-20
$X_o(B_i^j)$ [1]	22.8	50.8	64.4	77.9	86.1
$X(B_i^j)$ (4)	**45.5**	**65.8**	**74.0**	**80.9**	**87.1**

Results under unsupervised scenario: Table. 2 presents the CorLoc results obtained when we do not make use of any weak supervision provided by class labels. We consider entire YouTube Object Dataset as a whole and apply the proposed method on it. We basically rely upon kNN method to find good matching activators from other videos. We compare with other methods which reported such unsupervised results as well as other single video localization methods. It can be seen that our full-fledged method also achieves the best performance in such unsupervised scenario.

4.3 Evaluation on Bounding Box Filtering

In this subsection, we evaluate the effectiveness of the proposed bounding box filtering. We generate 300 bounding box proposals using [1] and select Top-k proposals based on either the objectness scores [1] or our confidence scores defined in Eq. (4). The recall rates are shown in Table 3. It can be seen that by incorporating the co-saliency based object prior for bounding box selection, our method greatly improves the recall rate. Even with only one proposal generated by our method, it has 45.5% probability to be overlapped with the ground truth bounding box with IoU large than 0.5, almost double of that in [1].

4.4 Evaluation on Co-saliency Prior and Tracklet Selection

In order to show the improvement in performance by using the developed co-saliency prior, we compare our method with an objectness based baseline, which is essentially our method but with top objectness-ranked bounding boxes using [1] instead of using our co-saliency prior. Table 4 shows the CorLoc results under different $m \in \{1, 3, 5, 10, 20\}$ in Table 4. It can be seen that our method achieves better overall CorLoc scores than the baseline for all m, which suggests that our co-saliency prior plays the key role here. When $m = 1$, the result signifies benefit

Table 4. Comparison with the objectness baseline with different m values.

	$m = 1$	$m = 3$	$m = 5$	$m = 10$	$m = 20$
Baseline	23.0	35.6	40.9	42.4	41.1
Proposed Method	**45.5**	**55.3**	**57.7**	**58.2**	**55.2**

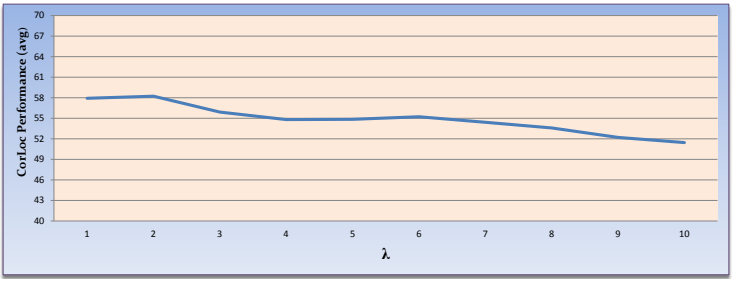

Fig. 7. Performance variation as λ in the Eq. (7) varies.

of co-saliency alone, which can be compared with the result obtained by Hough match alone in [14] (referring to the foreground saliency based on appearance only, i.e. F(A) at 1^{st} iteration. Kindly refer to [14] for more details). Ours is 45.5 compared to their 32. It can also be observed that as m increases, i.e. considering multiple candidate tracklets, the performance increases. This indicates that the co-saliency alone ($m = 1$ case) is not sufficient. Only when we combine the co-saliency prior with the tracklet generation and selection, we achieve the best performance. In the tracklet selction, we have the tradeoff parameter λ balancing the confidence and smoothness terms. In Fig.7, we show that when λ is set in range 1 to 6, performance varies between 55 and 58, which is somewhat stable. After $\lambda = 6$, performance drops because smoothness overweighs the confidence.

4.5 Limitations and Discussions

Although we consider objectness measure alongside with our object mask for selection of bounding boxes, incase co-saliency map based object prior is not good. In addition, we rely on the consistency of adjacent tracklets to negate the effect of few bad object priors. But it is quite possible that most of the activators fail in obtaining good co-saliency based object prior in a particular video. In such cases, proposed method is quite likely to fail. In Fig. 8, we show such failure examples of videos where most of the activators failed to obtain good co-saliency maps resulting in poor highest scored bounding box proposals.

Also, there are a few reasons for the relatively low performance of our method at some categories, as can be observed in Table 1. First, our method heavily relies on the co-saliency object prior. For some categories such as horse or motorbike, human beings often appear on horse or motorbike on several videos, which also

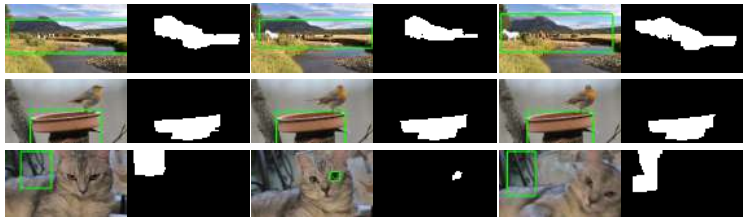

Fig. 8. Failure examples of videos where most of the activators failed to obtain good co-saliency based object prior (O_i) resulting in poor highest scored bounding box proposals

get highlighted in our co-saliency maps and included in our results, while they are excluded in the groundtruths of the two categories. Second, our parameters are all set globally instead of calibrated for individual categories. Thus, it is likely that for some other parameter setting, we might achieve better results. For example, in the case of bird category, if we select 8 bounding boxes instead of the default 10, we can improve the CorLoc result from 59.6% to 62.5%.

Execution Time: Our algorithm takes nearly 16 hours for co-localizing the entire YouTube Object dataset on PC with Intel Core i5-3470 (3.20 GHz, 4 cores) CPU. Whereas [14] takes 60 hours (from [14]) on PC with Xeon CPU (2.6 GHz, 12 cores). Therefore, our method is relatively faster.

5 Conclusion

We have proposed a new video co-localization method named *co-saliency activated tracklet selection* (CATS) where we activate several tracklets with the help of co-saliency maps at regular intervals. We then employ dynamic programming to select optimal tracklets from these sets for forming a tube to localize the common object. In contrast to previous methods, we proposed a guided single video-based framework which is non-iterative and computationally efficient. In the proposed approach, co-saliency plays the key role in guiding the activation and selection of our processing units called *co-saliency activated tracklets*, different from bounding box proposals or tube proposals used previously for the video co-localization problem. We obtain state-of-the-art localization results on YouTube Objects dataset in both weakly supervised and unsupervised scenarios through the proposed approach.

Acknowledgements

This research was carried out at the Rapid-Rich Object Search (ROSE) Lab at the Nanyang Technological University, Singapore. The ROSE Lab is supported by the National Research Foundation, Prime Ministers Office, Singapore, under its IDM Futures Funding Initiative and administered by the Interactive and Digital Media Programme Office. This work is supported in part by Singapore Ministry of Education Academic Research Fund Tier 2 MOE2015-T2-2-114.

References

1. Alexe, B., Deselaers, T., Ferrari, V.: Measuring the objectness of image windows. Transactions on Pattern Analysis and Machine Intelligence (T-PAMI), IEEE 34(11), 2189–2202 (Nov 2012)
2. Alt, N., Hinterstoisser, S., Navab, N.: Rapid selection of reliable templates for visual tracking. In: Computer Vision and Pattern Recognition (CVPR). pp. 1355–1362. IEEE (2010)
3. Bao, C., Wu, Y., Ling, H., Ji, H.: Real time robust l1 tracker using accelerated proximal gradient approach. In: Computer Vision and Pattern Recognition (CVPR). pp. 1830–1837. IEEE (2012)
4. Brox, T., Malik, J.: Object segmentation by long term analysis of point trajectories. In: European Conference on Computer Vision (ECCV), pp. 282–295. Springer (2010)
5. Chang, K.Y., Liu, T.L., Lai, S.H.: From co-saliency to co-segmentation: An efficient and fully unsupervised energy minimization model. In: Computer Vision and Pattern Recognition (CVPR). pp. 2129–2136. IEEE (2011)
6. Chen, H.T.: Preattentive co-saliency detection. In: International Conference on Image Processing (ICIP). pp. 1117–1120. IEEE (2010)
7. Fu, H., Cao, X., Tu, Z.: Cluster-based co-saliency detection. Transactions on Image Processing (T-IP), IEEE 22(10), 3766–3778 (2013)
8. Jacobs, D.E., Goldman, D.B., Shechtman, E.: Cosaliency: Where people look when comparing images. In: ACM symposium on User interface software and technology. pp. 219–228. ACM (2010)
9. Jain, M., Jegou, H., Bouthemy, P.: Better exploiting motion for better action recognition. In: Computer Vision and Pattern Recognition (CVPR). pp. 2555–2562. IEEE (2013)
10. Jerripothula, K.R., Cai, J., Yuan, J.: Group saliency propagation for large scale and quick image co-segmentation. In: International Conference on Image Processing (ICIP). pp. 4639–4643. IEEE (2015)
11. Jerripothula, K.R., Cai, J., Meng, F., Yuan, J.: Automatic image co-segmentation using geometric mean saliency. In: International Conference on Image Processing (ICIP). pp. 3282–3286. IEEE (2014)
12. Jiang, H., Wang, J., Yuan, Z., Wu, Y., Zheng, N., Li, S.: Salient object detection: A discriminative regional feature integration approach. In: Computer Vision and Pattern Recognition (CVPR). pp. 2083–2090. IEEE (2013)
13. Joulin, A., Tang, K., Fei-Fei, L.: Efficient image and video co-localization with frank-wolfe algorithm. In: European Conference on Computer vision (ECCV), pp. 253–268. Springer (2014)
14. Kwak, S., Cho, M., Laptev, I., Ponce, J., Schmid, C.: Unsupervised object discovery and tracking in video collections. In: International Conference on Computer Vision (ICCV). pp. 3173–3181. IEEE (2015)
15. Li, H., Ngan, K.N.: A co-saliency model of image pairs. Transactions on Image Processing (T-IP), IEEE 20(12), 3365–3375 (2011)
16. Liu, C., Yuen, J., Torralba, A.: Sift flow: Dense correspondence across scenes and its applications. Transactions on Pattern Analysis and Machine Intelligence (T-PAMI), IEEE 33(5), 978–994 (2011)
17. Liu, Z., Zou, W., Li, L., Shen, L., Le Meur, O.: Co-saliency detection based on hierarchical segmentation. Signal Processing Letters, IEEE 21(1), 88–92 (Jan 2014)

18. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: International Joint Conference on Artificial Intelligence (IJCAI). vol. 2, pp. 674–679. Morgan Kaufmann Publishers Inc. (1981)
19. Matthews, I., Ishikawa, T., Baker, S.: The template update problem. Transactions on Pattern Analysis and Machine Intelligence (T-PAMI), IEEE 26(6), 810–815 (2004)
20. Mei, X., Ling, H.: Robust visual tracking using l1 minimization. In: International Conference on Computer Vision (ICCV). pp. 1436–1443. IEEE (2009)
21. Mei, X., Ling, H., Wu, Y., Blasch, E., Bai, L.: Efficient minimum error bounded particle resampling l1 tracker with occlusion detection. Transactions on Image Processing (T-IP), IEEE 22(7), 2661–2675 (July 2013)
22. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. International journal of computer vision (IJCV), Springer 42(3), 145–175 (2001)
23. Papazoglou, A., Ferrari, V.: Fast object segmentation in unconstrained video. In: International Conference on Computer Vision (ICCV). pp. 1777–1784. IEEE (2013)
24. Prest, A., Leistner, C., Civera, J., Schmid, C., Ferrari, V.: Learning object class detectors from weakly annotated video. In: Computer Vision and Pattern Recognition (CVPR). pp. 3282–3289. IEEE (2012)
25. Rother, C., Kolmogorov, V., Blake, A.: "grabcut": Interactive foreground extraction using iterated graph cuts. In: SIGGRAPH 2004. pp. 309–314. ACM (2004)
26. Rubinstein, M., Joulin, A., Kopf, J., Liu, C.: Unsupervised joint object discovery and segmentation in internet images. In: Computer Vision and Pattern Recognition (CVPR). pp. 1939–1946. IEEE (2013)
27. Vicente, S., Rother, C., Kolmogorov, V.: Object cosegmentation. In: Computer Vision and Pattern Recognition (CVPR). pp. 2217–2224. IEEE (2011)
28. Wang, L., Hua, G., Sukthankar, R., Xue, J., Zheng, N.: Video object discovery and co-segmentation with extremely weak supervision. In: European Conference on Computer Vision (ECCV), pp. 640–655. Springer (2014)
29. Zhang, D., Han, J., Li, C., Wang, J.: Co-saliency detection via looking deep and wide. In: Computer Vision and Pattern Recognition. pp. 2994–3002. IEEE (2015)