Efficient video object co-localization with co-saliency activated tracklets

Jerripothula, Koteswar Rao; Cai, Jianfei; Yuan, Junsong

2018

Jerripothula, K. R., Cai, J., & Yuan, J. (2019). Efficient video object co-localization with co-saliency activated tracklets. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(3), 744-755. doi:10.1109/tcsvt.2018.2805811

<https://hdl.handle.net/10356/142175>

<https://doi.org/10.1109/TCSVT.2018.2805811>

© 2018 IEEE. All rights reserved.

Downloaded on 26 Apr 2023 15:31:52 SGT

Efficient Video Object Co-localization with Co-saliency Activated Tracklets

Koteswar Rao Jerripothula, *Member, IEEE*, Jianfei Cai, *Senior Member, IEEE*, and Junsong Yuan, *Senior Member, IEEE*

Abstract—Video object co-localization is the task of jointly localizing common visual objects across videos. Due to the large variations both across the videos and within each video, it is quite challenging to identify and track the common objects jointly. Unlike the previous joint frameworks that use a large number of bounding box proposals to attack the problem, we propose to leverage *co-saliency activated tracklets* to efficiently address the problem. To highlight the common object regions, we first explore inter-video commonness, intra-video commonness, and motion saliency to generate the co-saliency maps for a small number of selected key frames at regular intervals. Object proposals of high objectness and co-saliency scores in those frames are tracked across each interval to build tracklets. Finally, the best tube for a video is obtained through selecting the optimal tracklet from each interval with the help of confidence and smoothness constraints. Experimental results on the benchmark YouTube-Objects dataset show that the proposed method outperforms state-of-the-art methods in terms of accuracy and speed under both weakly supervised and unsupervised settings. Moreover, by noticing the existing benchmark dataset lacks of sufficient annotations for object localization (only one annotated frame per video), we further annotate more than 15k frames of the YouTube videos and develop a new benchmark dataset for video co-localization.

Index Terms—tracklets, video, co-localization, co-saliency.

I. INTRODUCTION

Localizing primary objects [1], [2] in videos is an important task in computer vision, since it facilitates many other vision tasks such as object recognition, retrieval and action recognition. Following the success of joint processing research in images [3], [4], [5], recent research interests have been shifted from single-video object localization to video object co-localization [6], [7], which aims at jointly localizing common objects across videos by exploiting shared attributes among videos as a type of weak supervision.

Video object 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

Fig. 1. Variations of common objects (i.e., cats) (a) across the videos as well as (b) within each video make the co-localization problem very challenging, while our method manages to localize the objects well.

viewpoint/pose changes (see Fig. 1(b)). Fourth, there are too many frames to deal with, which requires efficient algorithms.

A few video object co-localization works [6], [8] have been proposed in the literature. In particular, [6] proposed to co-select bounding box proposals, and [8] proposed 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 [9]. 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 [10], [11], [12], [6], [13]. This motivates us to propose a framework where we divide video object co-localization 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 individual video processing based framework. Similar to this idea, recently [7] developed a two-step framework for video object co-localization, where they iteratively discover common objects across neighboring videos and then incorporate the prior into individual video localization. However, [7] relies on matching large number of bounding box proposals extracted at every frame, which demands huge computational resources and memory.

Instead of relying on a large number of bounding box proposals, in this paper, we propose to leverage *co-saliency activated tracklets* for efficient video object co-localization. Usually, joint processing algorithms [6], [8], [9] succumb to expensive computations while processing a large number of

K. R. Jerripothula, J. Cai and J. Yuan are with Graphic Era University, Nanyang Technological University and State University of New York at Buffalo, respectively. E-mail: krjimp@geu.ac.in, asjfc@ntu.edu.sg and jsyuan@buffalo.edu. Contact author: Koteswar Rao Jerripothula.

Copyright © 2018 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending an email to pubs-permissions@ieee.org."

video frames together. Instead, we suggest joint processing of only a few sampled frames from the videos such that they can guide the remaining to improve the efficiency. 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 only a small number of uniformly sampled key frames. We then make use of the object prior to guide the selection of only a small set of proposals at each key frame and use them to activate the tracklets across the subsequent frames within each pair of adjacent key frames. Finally, guided by these key frames, we separately generate the best tube for each video by selecting the optimal tracklets based on confidence and consistency between adjacent tracklets. Experimental results on the benchmark YouTube-Objects dataset [8] show that our proposed method outperforms state-of-the-art methods in both accuracy and speed.

Note that our work is also motivated by the benefits of co-saliency and tracklets. Co-saliency research [14], [15], [16] has recently demonstrated significant contribution in object discovery problems. On the other hand, tracklets developed through trackers [17], [18], [19], [20], [21], [22], [23] are quite spatiotemporally consistent and reliable, especially for short video intervals. By converting the bounding box selection problem [6], [7] into a tracklet selection problem, we greatly reduce the complexity and also improve the temporal coherence.

We would like to point out that an early version of this work has been published in [24]. Compared with the conference version [24], this journal submission includes the following major extensions. 1) Considering the existing benchmark dataset, YouTube-Objects dataset, lacks of sufficient annotations for video object localization, we redevelop a benchmark dataset, named YouTube Co-localization dataset, specifically for video co-localization. 2) We provide detailed evaluations of our method on the performance of neighborhood discovery. 3) We also give more detailed explanations or demonstrations on our method.

In summary, the main contributions of this paper are threefold: 1) exploring inter-video, intra-video and motion information to generate a co-saliency based object prior for activating tracklets; 2) leveraging tracklets for video object co-localization; and 3) developing a new video object co-localization benchmark dataset.

II. RELATED WORK

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

A. Video Co-localization

Video co-localization is a task of jointly localizing the shared object in a set of videos. The recent work of [6] and [8] proposed joint frameworks to locate common objects across videos. In [6], it used Quadratic Programming framework to co-select bounding box proposals in all the frames in all the videos together. While in [8], 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 a difficult task for such joint frameworks, especially when each bounding box in a frame or candidate tube for an entire video is chosen as a processing unit. In contrast, our method is based on individual video processing while with the guidance of the co-saliency based object prior that incorporates the information from other related videos. Recently, [7] proposed an approach of developing foreground confidence for bounding boxes and selecting bounding boxes while maintaining temporal consistency. The presence of noisy bounding box proposals mandates taking an iterative approach in [7]. On the contrary, our method leverages on tracklets instead of bounding boxes, which significantly reduces the computational complexity. In addition, our developed co-saliency based object prior can effectively eliminate noisy bounding box proposals, which facilitates our method to be non-iterative and thus more efficient.

There are also some other video co-localization works. For example, [15] overcomes the assumption made in [6], [7], [8], which assume the common object is present at all the frames of all videos, by providing a few labels of relevant frames and irrelevant frames to guide object discovery effectively. Also, with the help of the pre-trained deep learning network and motion cues, [25] reported some improved co-localization results on videos. However, it is limited to the categories that the pre-trained network has seen.

B. 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 [26], 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 [27], [28]. In [29], 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 [30], which learns the global correspondence and obtains cluster saliency quite well. However, [30] was mainly designed for images of the same (or very similar) object instances captured at different viewpoints or time. Therefore, it struggles to handle image sets with huge intra-class variation. There have been some fusion approaches [31], [32], [33], [13] as well where raw saliency maps are fused to generate co-saliency maps. For example, [31] fused not only the saliency maps of the same image generated by different saliency detection methods but also the saliency maps from different images such that common object regions get emphasized in the co-saliency maps. In [14], it introduced deep intra-group semantic information and wide cross-group heterogeneous 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.

Fig. 2. Overview of the proposed *co-saliency activated tracklets based* efficient video object 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 the same color across a video trunk constitute a tracklet.

III. 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, 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.

A. Co-saliency Based Object Prior Generation

To generate good object prior, our basic idea is to combine the following three types 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 due to camera/object motion or scene change, 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 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 $\mathcal{A} = \{A_1, A_2, \dots, A_n\}$ be a set of n activators (sampled uniformly) 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 an externally matched activator set $\mathcal{N}_i^{ext} = \{\mathbb{A} | \delta(A_i, \mathbb{A}) < \epsilon, \mathbb{A} \in \mathbb{V} \setminus \mathcal{V}\}$, where $\mathbb{A}$ denotes an externally matched activator and δ denotes a distance function. Particularly, we extract the GIST descriptor [34] from each activator weighted by its initial saliency map [35]. We employ the strategy used by [12] for weighing the GIST descriptor. The Gabor filter responses are multiplied by a given weight mask. It gives higher weights to foreground pixels when the responses are averaged over image blocks while computing the descriptor. The distance $\delta(A_i, \mathbb{A})$ between a pair of activators

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.

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

$$M_i^{ext} = \frac{\mathcal{S}(A_i) + \sum_{A \in \mathcal{N}_i^{ext}} \mathcal{W}_A^{A_i}(\mathcal{S}(A))}{|\mathcal{N}_i^{ext}| + 1} \quad (1)$$

where $\mathcal{S}(\cdot)$ denotes the initial saliency map filter, $\mathcal{W}_A^{A_i}(\cdot)$ denotes the warping function from A to A_i , and $|\cdot|$ denotes cardinality. We use the masked dense SIFT correspondence (SIFT flow) [36], [12] to find pixel correspondences for the warping. Eq. (1) essentially computes the joint saliency of the matched object points in different activators by averaging their saliency values.

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 the weighted GIST descriptor as discussed before. Then, for an activator, other activators in its cluster are considered as its matches. Therefore, internally matched activator set $\mathcal{N}_i^{int} = \{A_j | A_j \in C_k \setminus A_i, A_i \in C_k\}$ is basically all other activators in cluster C_k to which A_i belongs. 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.

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

where the definitions of $\mathcal{S}$ and $\mathcal{W}$ remain same as those defined previously. Here, when 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.

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 ω -flow method in [37] to extract the motion saliency map for each frame in a video trunk. Considering that for deformable objects, parts of the object could move while other parts might remain still (see

Fig. 4. Background saliency defects present in visual saliency may not be present in the motion saliency map. Hence when they are fused, such background defects get subdued.

Fig. 3 for example). Thus, we propose to use max pooling to collect motion saliency from an activator and its subsequent frames after warping, which we call *motion co-saliency* M_i^{mot} , defined as

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

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 subsequent 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. Such fusion of co-saliency maps from different sources also helps subdue possible saliency defects existing in individual maps. The defects could be salient background elements persisting in the inter-video co-saliency, or background saliency in the motion co-saliency map due to too much object deformation. These defects may often exist in individual maps, but not in all the maps. For example, visually salient background for the baby image (see Fig. 4) may result in the salient background in the inter-video co-saliency map as well, but it is not there in the motion co-saliency map (see Fig. 3). Once the final fused co-saliency map is available (see Fig. 5 for examples), we apply the GrabCut [38] to obtain a binary segmentation mask, denoted as object prior O_i , for activator A_i . Specifically, there are two thresholds for generating the segmentation seeds for GrabCut: one for obtaining background seeds and the other one for obtaining foreground seeds. We use Otsu's method to obtain the threshold for background seeds and 0.7 as the one for foreground seeds. Otsu's threshold is well known for its robustness, the threshold of 0.7 is based on our empirical observations. In addition, since we are only looking for a rough segmentation (as the goal is localization), one iteration of Grabcut proves to be sufficient.

B. 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 [9], [7], we also use bottom-up object proposal techniques, particularly [39], to generate initial bounding boxes. However, to ensure a high

Fig. 5. Final fused co-saliency maps for some activator samples in YouTube-Object dataset

object detection rate, the existing general object proposal techniques typically need 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 [39] 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

$$X(B_i^j) = X_o(B_i^j) + J(B_i^j, B_i^o) \quad (4)$$

where $X_o(B_i^j)$ is the objectness score (between 0 and 1) directly obtained from [39] and $J(\cdot)$ is Jaccard similarity function (also called IoU, intersection over union). We then select the top- m proposals with highest object confidence scores.

Tracklet confidence scores: Once the top- m candidate bounding box proposals are selected at the activator, tracklets are obtained using the existing tracker [22] starting from these m 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:

$$X_f(T_i^j) = J(B_i^j, B_i^o), \quad (5)$$

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

where X_f and X_l are the first and the last confidence scores of a tracklet based on our object priors at its two ends,

respectively. Since we do not have objectness score (X_o) for the last bounding box produced by tracking, we omit the use of objectness score here, even for the first bounding box. Thus, these scores rely completely on co-saliency.

C. 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 spatiotemporal consistent tube which localizes the common object with high confidence. Let $\mathcal{T} = \{T_1, T_2, \dots, T_n\}$ be a possible tube. Our goal is to find the best tube for every video that minimizes the following criterion, i.e.

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

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_{i+1}^o . 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. Essentially, the first term signifies the reliance on co-saliency, and the second term signifies the reliance on temporal consistency between adjacently activated tracklets. In this way, selecting the optimal tracklets results in co-localizing the common video object. This problem of Eq. (7) can be easily solved using dynamic programming or shortest path algorithm.

Fig. 6. Sample localization results (red) along with groundtruths (green) on YouTube-Objects dataset.

TABLE I
CORLOC RESULTS OF VIDEO OBJECT CO-LOCALIZATION ON YOUTUBE-OBJECTS DATASET UNDER WEAKLY SUPERVISED SCENARIOS.

	aeroplane	bird	boat	car	cat	cow	dog	horse	motorbike	train	avg
[8]	51.7	17.5	34.4	34.7	22.3	17.9	13.5	26.7	41.2	25.0	28.5
[6]	25.1	31.2	27.8	38.5	41.2	28.4	33.9	35.6	23.1	25.0	31.0
[7]	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
adv. tools	76.8	68.3	66.7	64.4	50.4	66.9	69.5	57.4	41.0	38.8	60.0

IV. EXPERIMENTAL RESULTS

We evaluate our method on the benchmark YouTube-Objects 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-Objects 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 object co-localization. The videos are sparsely annotated, and even for those annotated videos, only one frame is annotated. The total number of bounding box annotations is around 1.4k, which is insufficient for the video co-localization problem. Therefore, we create a new dataset, named YouTube Co-localization dataset¹, annotating more than 15k frames of these downloaded YouTube videos.

A. Implementation Details

Unless specified, activators are chosen at the interval of 50 frames in any experiment. While calculating inter-video co-

saliency, we use kNN to ensure there are at least 10 best-matched activators available from other videos. 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 [35] to generate saliency maps for individual activators. We choose $m = 10$ at the bounding-box filtering step, and sample every 5th frame between activators to generate the total motion saliency map for the preceding activator to avoid repetitiveness. The parameter λ introduced in Eq. (7), i.e. weight for temporal consistency, is set to 2, same as [7]. For the off-the-shelf techniques, including tracklets [22], motion saliency [37] and GrabCut [38], we use their default settings.

B. Performance Comparison

Results under weakly supervised scenarios: Table I shows the CorLoc performance on YouTube Object dataset under weakly supervised scenarios using our full-fledged 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 object co-localization. It can be seen that we almost double the average performance of [8] and [6] that simultaneously locate the

¹Available online at <https://sites.google.com/site/koteswarraojerripothula/>

Fig. 7. Our video object co-localization results on YouTube-Objects dataset. 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 II
CORLOC RESULTS ON YOUTUBE-OBJECTS DATASET IN AN UNSUPERVISED SCENARIO WHERE WE DO NOT USE CLASS LABELS.

	aeroplane	bird	boat	car	cat	cow	dog	horse	motorbike	train	avg
[40]	53.9	19.6	38.2	37.8	32.2	21.8	27.0	34.7	45.4	37.5	34.8
[9]	65.4	67.3	38.9	65.2	46.3	40.2	65.3	48.4	39.0	25.0	50.1
[7]	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
adv. tools	75.2	67.3	66.7	62.5	50.4	66.9	67.4	56.6	36.0	40.5	58.9

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 together, 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% relative improvement for the average performance over the state-of-the-art [7], 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, [7] 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). Note that since [7] included frames with groundtruths while sampling, we also do the same while sampling our activators for a fair comparison. Considering that more noisy nodes are eliminated, it results in more reliable results. In addition, [7] 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.

In addition to our full-fledged method (*ext+int+mot*), in Table I, we also show the results of the variants (*ext*, *int+mot*) that use different combinations of the co-saliency maps. The

TABLE III

NEIGHBORHOOD RETRIEVAL PERFORMANCE COMPARISON. THE HIGHER THE CORRET VALUE THE BETTER IT IS, WHEREAS THE LOWER THE TOP-X ERROR RATE THE BETTER IT IS.

	CorRet	Top-1 error rate	Top-2 error rate
[7]	35.5	51.3	34.0
$\mathcal{N}_i^{ext}$	42.7	40.2	25.1
Ours	51.4	33.4	20.8

TABLE IV

CATEGORY RELATIONSHIPS OF YOUTUBE-OBJECTS DATASET OBTAINED BY OUR METHOD

Category	Related to	Reasons
Aeroplane	Boat	Vehicles
Bird	Aeroplane and Dog	Flight
Boat	Aeroplane	Vehicles
Car	Aeroplane, Boat and Horse	Vehicles
Cat	Dog and Horse	Animals
Cow	Horse	Animals
Dog	Cat, Cow and Horse	Animals
Horse	Cat, Dog and Cow	Animals
Motorbike	Horse and Cow	Ride
Train	-	-

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.

Moreover, since our method is not limited to a particular set of tools, we employ advanced (adv.) tools to improve our accuracy and efficiency. In particular, we employ the scheme used in [41] for matching activators with dense SIFT to replace GIST. As for intra-video co-saliency, we employ colorSIFT instead of combining color and SIFT into one descriptor. Also, noting that generating DRFI [35] saliency and w-flow saliency is slow, we use MBD [42] for generating saliency and also apply it on variance maps of subsequent frames for generating the motion saliency. Lastly, we also adopt selective search scheme for generating object proposals. By incorporating all these advanced tools, we now achieve 8% relative improvement over [7].

In Fig. 6, 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 primary objects with various poses and shapes across the videos. In Fig. 7, 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 the dog video. At the same time, our method is also able to handle objects that do not move much such as in the bird's video. These results clearly demonstrate the robustness of our method.

Results under unsupervised scenario: Table II presents the CorLoc results obtained when we do not make use of any weak supervision provided by class labels. We consider entire YouTube-Objects dataset as a whole and apply the proposed method to it. We basically rely on 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 proposed method also achieves the best performance in such unsupervised scenario.

Following [7], we also evaluate our neighborhood retrieval performance and compare with [7]. Here, same as [7], we use CorRet and Top-k error rates as the performance metrics, where CorRet refers to the percentage of frames retrieved having the same label as the target video, and Top-k error rate refers to the percentage of videos whose labels do not fall among the k most frequent labels of their neighbor frames. It turns out that even our initial neighborhood $\mathcal{N}_i^{ext}$ obtained using the original saliency map outperforms [7]'s final neighborhood retrieval performance, as observed in Table III. This explains why our method does not require any iterations, i.e. we have good neighborhood right at the beginning. Furthermore, when we use our final fused co-saliency map achieved in the unsupervised setting for the retrieval, the results become even better. We also compare our confusion matrix with that of [7] in Fig. 8. In the confusion matrix, for videos with their groundtruth labels on the left, we compute the composition of their neighbors' labels. So, the diagonal elements represent CorRet values. It can be seen that our confusion matrix is much better than that of [7], where for our method the largest portion of the retrieved neighborhood frames for each class are always from the same class, whereas [7] misses out in the Cat and Motorbike classes. In addition, Table IV summarizes the category relationships that our method discovered, where for each category we list all other categories that occupy at least 10% of its retrieved neighbors. It can be seen that most of the related categories are semantically meaningful (in green) to the concerned category except the three instances in red.

C. YouTube Co-localization Dataset

Similar to YouTube-Objects dataset, our redeveloped YouTube Co-localization dataset also consists of 10 categories, with only one change: we turn motorbike category into biker category. This is because, in almost every video of motorbike category, there is a person sitting on the motorbike. In the context of video object co-localization, both the person and the motorbike should be counted as the common object, instead of only counting the motorbike in the original object dataset. Therefore, in our biker category, bounding box annotations enclose both the person and the motorbike region. Note that although some of the horse videos also have persons, it is not necessary to include the person since it does not happen in every video. The entire dataset consists of total 1,198 videos with a total of 15,345 frames annotated (averagely 13 annotated frames per video). Sample annotations are shown in the Fig. 9. Table V gives the detailed results on our newly developed YouTube Co-localization dataset for benchmarking. We show average performances at three levels: category, video and frame. Nearly similar performances (nearly 54 CorLoc and 48 IoU) at all three levels show that our method is quite reliable.

Fig. 8. Confusion matrix comparison between [7] and our method. Note how our method produces maximum at diagonal for all horizontal target labels, as it would be expected in neighborhood retrieval, whereas [7] misses out for Cat and Motorbike categories.

Fig. 9. Sample annotations from our YouTube Co-localization dataset.

TABLE V
DETAILED RESULTS ON OUR YOUTUBE CO-LOCALIZATION DATASET

	plane	bird	boat	car	cat	cow	dog	horse	biker	train	avg (category)	avg (video)	avg (frame)
No. of videos	296	102	113	107	123	70	120	61	93	113	-	-	-
No. of annotations	2,103	1,519	1,462	810	2,833	1,017	1,385	1,156	931	2,109	-	-	-
Weakly Supervised (CorLoc)	61.5	52.5	64.3	62.5	47.3	69.9	55.6	53.9	45.5	34.3	54.7	55.4	53.5
Weakly Supervised (IoU)	52.4	45.8	52.6	51.5	45.0	57.7	48.5	48.8	43.4	39.6	48.3	48.5	47.9
Unsupervised (CorLoc)	59.1	53.0	61.5	58.9	47.5	70.6	56.3	49.8	45.0	37.7	53.9	54.4	53.4
Unsupervised (IoU)	51.2	46.0	50.8	50.2	45.6	57.9	48.7	48.1	42.1	40.9	48.2	48.4	47.9

TABLE VI

THE RECALL PERFORMANCE ON YOUTUBE-OBJECTS DATASET USING EITHER THE EXISTING OBJECTNESS SCORES $X_o(B_i^j)$ [39] OR THE PROPOSED OBJECT CONFIDENCE SCORES $X(B_i^j)$ IN EQ. (4) FOR BOUNDING BOX SELECTION.

	Top-1	Top-3	Top-5	Top-10	Top-20
$X_o(B_i^j)$ [39]	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

TABLE VII

COMPARISON OF THE CORLOC RESULTS WITH THE OBJECTNESS BASELINE WITH DIFFERENT m VALUES (NO. OF TRACKLETS GENERATED).

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

D. 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 [39] and select Top-k proposals based on either the objectness scores [39] or our confidence scores defined in Eq. (4). The recall rates are shown in Table VI. 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 being selected by our method, it has 45.5% probability to be overlapped with the ground truth bounding box with IoU larger than 0.5, almost double of that in [39].

E. Evaluation on Co-saliency Prior and Tracklet Selection

In order to show the contribution of the developed co-saliency prior to the performance improvement, we compare our method with an objectness based baseline, which is essentially our method but with top objectness-ranked bounding boxes using [39] instead of using our co-saliency prior. The top objectness-ranked bounding box proposal is chosen as B_i^0 in the baseline approach. We compare them while varying the number of tracklets generated, i.e. m . Table VII shows the CorLoc results under different $m \in \{1, 3, 5, 10, 20\}$ in Table VII. It can be seen that our method achieves much 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 the benefit of co-saliency alone, which can be compared with the result obtained by

Fig. 10. Performance variation as λ in the Eq. (7) varies. In the range 1-6, the performance remains stable. However, after $\lambda = 6$, the performance drops because the smoothness significantly over-weighs the confidence.

Hough match alone in [7] (referring to the foreground saliency based on appearance only in [7]). 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 selection, we achieve the best performance. In the tracklet selection, we have the trade-off parameter λ in Eq. (7) balancing the confidence and smoothness terms. Fig.10 shows that when λ is set in range 1 to 6, the performance varies between 55 and 58, which is stable. After $\lambda = 6$, the performance drops because the smoothness significantly over-weighs the confidence.

F. Limitations and Discussions

Our method heavily relies on the co-saliency based object prior. Thus, if the co-saliency based object prior is not good in the most activators of a video, the proposed method is likely to fail. Fig. 11 shows some failure examples of videos where most of the activators failed to obtain good co-saliency maps resulting in poorly selected bounding box proposals. Possible reasons include: (i) poor correspondences with the identified neighbors; (ii) the existence of some other very salient or primary objects; and (iii) the lack of good neighbors.

Also, our method obtains relatively low performance in some categories, as can be observed in Table I. First, as

Fig. 11. Failure examples of videos where most of the activators failed to obtain good co-saliency based object prior (O_i) resulting in wrongly selected bounding box proposals.

TABLE VIII
RUNTIME BREAKUP OF PROPOSED METHOD ON YOUTUBE OBJECTS DATASET

Components	Co-saliency Generation	Tracklets Generation	Tube Generation	End-to-End
Time Taken (hrs)	8.2	3.8	0.9	12.9

mentioned, for some categories such as motorbike, human beings often appear on motorbike on most videos, which also get highlighted in our co-saliency maps and included in our results, while they are excluded in the ground-truths of YouTube-Objects dataset. Second, our parameters are all set globally instead of being calibrated for individual categories. Thus, it is likely that for some other parameter setting, we might achieve better results. For example, in few categories, when we choose different number of bounding boxes, we achieve better performance.

Execution Time: Our algorithm nearly takes 13 hours (with advanced tools) for co-localizing the YouTube-Objects dataset on PC with Xeon CPU (3.40 GHz, 4 cores) using Matlab, whereas [7] reports taking 60 hours on PC with Xeon CPU (2.6 GHz, 12 cores). The runtime breakup is shown in the Table VIII.

V. CONCLUSION

We have proposed an efficient video object co-localization method using *co-saliency activated tracklets*. In the proposed method, we activate several tracklets with the help of co-saliency maps at regular intervals. We then select the optimal tracklets to form a tube to localize the common object. We have proposed a guided individual video processing based framework which is non-iterative and computationally efficient. Instead of relying on a large number of bounding box proposals, we leverage *co-saliency activated tracklets* for video co-localization. In our approach, co-saliency plays the key role in selecting the activating bounding boxes (for tracklets) and the tracklets (for the final tube). Our method obtains state-of-the-art localization results on YouTube-Objects dataset in both weakly supervised and unsupervised scenarios. Moreover, considering the lack of sufficient annotations in the current benchmark (YouTube-Objects dataset), we further annotate more than 15k frames to create a new dataset called YouTube Co-localization dataset, specifically for the video object co-localization benchmarking purpose.

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 Infocomm Media Development Authority, Singapore.

REFERENCES

- [1] Y. Luo, J. Yuan, P. Xue, and Q. Tian, "Saliency density maximization for efficient visual objects discovery," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 12, pp. 1822–1834, Dec 2011.
- [2] J. Yang, G. Zhao, J. Yuan, X. Shen, Z. Lin, B. Price, and J. Brandt, "Discovering primary objects in videos by saliency fusion and iterative appearance estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 6, pp. 1070–1083, June 2016.
- [3] F. Meng, J. Cai, and H. Li, "On multiple image group cosegmentation," in *Asian Conference on Computer Vision (ACCV)*. Springer, 2014, pp. 258–272.
- [4] F. Meng, H. Li, S. Zhu, B. Luo, C. Huang, B. Zeng, and M. Gabbouj, "Constrained directed graph clustering and segmentation propagation for multiple foregrounds cosegmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 11, pp. 1735–1748, Nov 2015.
- [5] K. R. Jeripothula, J. Cai, J. Lu, and J. Yuan, "Object co-skeletonization with co-segmentation," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 3881–3889.
- [6] A. Joulin, K. Tang, and L. Fei-Fei, "Efficient image and video co-localization with frank-wolfe algorithm," in *European Conference on Computer vision (ECCV)*. Springer, 2014, pp. 253–268.
- [7] S. Kwak, M. Cho, I. Laptev, J. Ponce, and C. Schmid, "Unsupervised object discovery and tracking in video collections," in *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 2015, pp. 3173–3181.
- [8] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari, "Learning object class detectors from weakly annotated video," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 3282–3289.
- [9] A. Papazoglou and V. Ferrari, "Fast object segmentation in unconstrained video," in *International Conference on Computer Vision (ICCV)*. IEEE, 2013, pp. 1777–1784.
- [10] K. R. Jeripothula, J. Cai, and J. Yuan, "Quality-guided fusion-based co-saliency estimation for image co-segmentation and co-localization," *IEEE Transactions on Multimedia*, vol. PP, no. 99, pp. 1–1, 2018.
- [11] S. Vicente, C. Rother, and V. Kolmogorov, "Object cosegmentation," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011, pp. 2217–2224.
- [12] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu, "Unsupervised joint object discovery and segmentation in internet images," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2013, pp. 1939–1946.
- [13] K. R. Jeripothula, J. Cai, and J. Yuan, "Qcse: Quality constrained co-saliency estimation for common object detection," in *Visual Communications and Image Processing (VCIP)*. IEEE, 2015, pp. 1–4.
- [14] D. Zhang, J. Han, C. Li, and J. Wang, "Co-saliency detection via looking deep and wide," in *Computer Vision and Pattern Recognition*. IEEE, 2015, pp. 2994–3002.
- [15] L. Wang, G. Hua, R. Sukthankar, J. Xue, and N. Zheng, "Video object discovery and co-segmentation with extremely weak supervision," in *European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 640–655.
- [16] Z. Liu, W. Zou, L. Li, L. Shen, and O. Le Meur, "Co-saliency detection based on hierarchical segmentation," *Signal Processing Letters, IEEE*, vol. 21, no. 1, pp. 88–92, Jan 2014.
- [17] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, ser. IJCAI'81. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1981, pp. 674–679.
- [18] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, vol. 26, no. 6, pp. 810–815, 2004.
- [19] N. Alt, S. Hinterstoisser, and N. Navab, "Rapid selection of reliable templates for visual tracking," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 1355–1362.

- [20] X. Mei and H. Ling, "Robust visual tracking using l1 minimization," in *International Conference on Computer Vision (ICCV)*. IEEE, 2009, pp. 1436–1443.
- [21] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai, "Efficient minimum error bounded particle resampling l1 tracker with occlusion detection," *Transactions on Image Processing (T-IP)*, vol. 22, no. 7, pp. 2661–2675, July 2013.
- [22] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust l1 tracker using accelerated proximal gradient approach," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 1830–1837.
- [23] Q. Chen, Q. S. Sun, P. A. Heng, and D. S. Xia, "Two-stage object tracking method based on kernel and active contour," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 4, pp. 605–609, April 2010.
- [24] K. R. Jeripothula, J. Cai, and J. Yuan, "Cats: Co-saliency activated tracklet selection for video co-localization," in *European Conference on Computer vision (ECCV)*. Springer, 2016, pp. 187–202.
- [25] P. Tokmakov, K. Alahari, and C. Schmid, *Weakly-Supervised Semantic Segmentation Using Motion Cues*. Cham: Springer International Publishing, 2016, pp. 388–404.
- [26] D. E. Jacobs, D. B. Goldman, and E. Shechtman, "Cosaliency: Where people look when comparing images," in *ACM symposium on User interface software and technology*. ACM, 2010, pp. 219–228.
- [27] H.-T. Chen, "Preattentive co-saliency detection," in *International Conference on Image Processing (ICIP)*. IEEE, 2010, pp. 1117–1120.
- [28] H. Li and K. N. Ngan, "A co-saliency model of image pairs," *Transactions on Image Processing (T-IP)*, vol. 20, no. 12, pp. 3365–3375, 2011.
- [29] 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 *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011, pp. 2129–2136.
- [30] H. Fu, X. Cao, and Z. Tu, "Cluster-based co-saliency detection," *Transactions on Image Processing (T-IP)*, vol. 22, no. 10, pp. 3766–3778, 2013.
- [31] K. R. Jeripothula, J. Cai, and J. Yuan, "Image co-segmentation via saliency co-fusion," *IEEE Transactions on Multimedia*, vol. 18, no. 9, pp. 1896–1909, Sept 2016.
- [32] K. R. Jeripothula, J. Cai, F. Meng, and J. Yuan, "Automatic image co-segmentation using geometric mean saliency," in *International Conference on Image Processing (ICIP)*. IEEE, 2014, pp. 3282–3286.
- [33] K. R. Jeripothula, J. Cai, and J. Yuan, "Group saliency propagation for large scale and quick image co-segmentation," in *International Conference on Image Processing (ICIP)*. IEEE, 2015, pp. 4639–4643.
- [34] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International journal of computer vision(IJCV)*, vol. 42, no. 3, pp. 145–175, 2001.
- [35] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: A discriminative regional feature integration approach," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2013, pp. 2083–2090.
- [36] C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications," *Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, vol. 33, no. 5, pp. 978–994, 2011.
- [37] M. Jain, H. Jegou, and P. Bouthemy, "Better exploiting motion for better action recognition," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2013, pp. 2555–2562.
- [38] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *Transactions on Graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 309–314.
- [39] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, vol. 34, no. 11, pp. 2189–2202, Nov 2012.
- [40] T. Brox and J. Malik, "Object segmentation by long term analysis of point trajectories," in *European Conference on Computer Vision (ECCV)*. Springer, 2010, pp. 282–295.
- [41] K. Tang, A. Joulin, L.-J. Li, and L. Fei-Fei, "Co-localization in real-world images," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014, pp. 1464–1471.
- [42] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Mch, "Minimum barrier salient object detection at 80 fps," in *IEEE International Conference on Computer Vision(ICCV)*, 2015.

Koteswar Rao Jeripothula (S'15-M'17) received his Ph.D. degree from Nanyang Technological University (NTU), Singapore, in 2017. Before that, he received his B.Tech degree from IIT Roorkee, India, in 2012, and he interned with Visual Modeling & Analytics Team in ADSC, Singapore, during the July-Dec'16 period. He is a faculty with Graphic Era University (GEU), India. His research interests include computer vision and multimedia. He has received "Top 10% paper" award at IEEE ICIP'14 and published in several top venues like IEEE CVPR, ECCV, IEEE T-MM, IEEE T-CSVT, etc.

Jianfei Cai (S'98-M'02-SM'07) received his PhD degree from the University of Missouri-Columbia. He is a faculty with School of Computer Engineering, Nanyang Technological University, Singapore. He has served as the Head of Visual & Interactive Computing Division and the Head of Computer Communication Division at NTU. His major research interests include computer vision, multimedia and machine learning. He has published more than 200 technical papers in international conferences and journals. He is a co-recipient of paper awards in ACCV, IEEE ICIP and MMSP. He has served as the leading Technical Program Chair for IEEE International Conference on Multimedia & Expo (ICME) 2012 and the leading General Chair for Pacific-rim Conference on Multimedia (PCM) 2012. He is currently an Associate Editor for IEEE Trans on Multimedia (TMM) and has served as an Associate Editor for IEEE Trans on Image Processing (T-IP) during 2013-2017 and for IEEE Trans on Circuits and Systems for Video Technology (T-CSVT) during 2007-2013. He also serves as the Chair for IEEE CAS Visual Signal Processing and Communication Technical Committee (VSPC-TC).

Junsong Yuan (M'08-SM'14) received his Ph.D. from Northwestern University and M.Eng. from National University of Singapore. Before that, he graduated from the Special Class for the Gifted Young of Huazhong University of Science and Technology, Wuhan, China, in 2002.

He is currently an associate professor at CSE department of State University of New York at Buffalo. Before that, he was an associate professor at Nanyang Technological University (NTU), Singapore. His research interests include computer vision, video analytics, gesture and action analysis, large-scale visual search and mining. He received best paper award from Intl. Conf. on Advanced Robotics (ICAR'17), 2016 Best Paper Award from IEEE Trans. on Multimedia, Doctoral Spotlight Award from IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'09), Nanyang Assistant Professorship from NTU, and Outstanding EECS Ph.D. Thesis award from Northwestern University.

He is currently Senior Area Editor of Journal of Visual Communications and Image Representations (JVCI), Associate Editor of IEEE Trans. on Image Processing (T-IP), IEEE Trans. on Circuits and Systems for Video Technology (T-CSVT), and served as Guest Editor of International Journal of Computer Vision (IJCV). He is Program Co-chair of ICME'18 and VCIP'15, and Area Chair of ACM MM'18, ICPR'18, CVPR'17, ICIP'17, ACCV'18'14 etc.