Unsupervised Object Discovery and Tracking in Video Collections

Suha Kwak[1,*] Minsu Cho[1,*] Ivan Laptev[1,*] Jean Ponce[2,*] Cordelia Schmid[1,†]

[1]Inria [2]École Normale Supérieure / PSL Research University

Abstract

This paper addresses the problem of automatically localizing dominant objects as spatio-temporal tubes in a noisy collection of videos with minimal or even no supervision. We formulate the problem as a combination of two complementary processes: discovery and tracking. The first one establishes correspondences between prominent regions across videos, and the second one associates similar object regions within the same video. Interestingly, our algorithm also discovers the implicit topology of frames associated with instances of the same object class across different videos, a role normally left to supervisory information in the form of class labels in conventional image and video understanding methods. Indeed, as demonstrated by our experiments, our method can handle video collections featuring multiple object classes, and substantially outperforms the state of the art in colocalization, even though it tackles a broader problem with much less supervision.

1. Introduction

Visual learning and interpretation usually have been formulated as a supervised classification problem, with manually selected bounding boxes acting as (strong) supervisory signal [8, 10]. To reduce human effort and subjective biases in manual annotation, recent work has addressed the discovery and localization of objects from weakly-annotated or even unlabelled datasets [4, 5, 9, 30, 32]. However, this task is difficult, and most approaches today still lag significantly behind strongly-supervised methods. With the ever growing popularity of video sharing sites such as YouTube, recent research has started to address similar problems in videos [17, 27, 29, 38], and has shown that exploiting the space-time structure of the world, which is absent in static images (*e.g*., motion information), may be crucial for achieving object discovery or localization with less supervision.

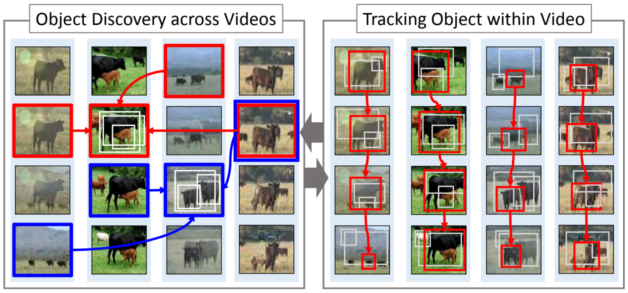


Figure 1. Given a noisy collection of videos, dominant objects are automatically localized as spatio-temporal "tubes". A discovery process establishes correspondences between prominent regions across videos (left), and a tracking process associates similar object regions within the same video (right). (Best viewed in color.)

Concretely, this paper addresses the problem of spatio-temporal object localization in videos with minimal supervision or even no supervision. Given a noisy collection of videos with multiple object classes, dominant objects are identified as spatio-temporal "tubes" (see definition in Section 1.2) for each video. We formulate the problem as a combination of two complementary processes: object *discovery* and *tracking* (Fig. 1). Object discovery establishes correspondences between regions depicting similar objects in frames of different videos, and object tracking temporally associates prominent regions within individual videos. Better object discovery enhances tracking, which in turn corrects erroneous discovery results and improves the correspondences across videos. Building upon recent advances in efficient matching [4] and tracking [26], we combine region matching across different videos and region tracking within each video into a joint optimization framework. We demonstrate that the proposed method substantially outperforms the state of the art in colocalization [17] on the YouTube-Object dataset, even though it tackles a broader problem with much less supervision.

1.1. Related work

Our approach combines object discovery and tracking. The discovery part establishes correspondences between

[*] WILLOW project-team, Département d'Informatique de l'Ecole Normale Supérieure, ENS/Inria/CNRS UMR 8548.

[†] LEAR project-team, Inria Grenoble Rhône-Alpes, Laboratoire Jean Kuntzmann, CNRS, Univ. Grenoble Alpes, France.

frames across videos to detect object candidates. Related approaches have been proposed for salient region detection [18], image cosegmentation [36, 37], and image colocalization [4]. Conventional object tracking methods [41] usually require annotations for at least one frame [13, 15, 40], or object detectors trained for target classes in a supervised manner [1, 2, 26]. Our method does not require such supervision and instead alternates discovery and tracking of object candidates.

The problem we address is closely related to video object colocalization [17, 27], whose goal is to localize common objects in a video collection. Prest *et al.* [27] generate spatio-temporal tubes of object candidates, and select one of these per video through energy minimization. Since the candidate tubes rely only on clusters of point tracks [3], this approach is not robust against noisy tracks and incomplete clusters. Joulin *et al.* [17] extend the image colocalization framework [32] to videos using an efficient optimization approach. Their method does not explicitly consider correspondences between frames from different videos, which are shown to be critical for robust localization of common objects by our experiments (Section 5.3).

Our setting is also related to object segmentation or cosegmentation in videos. For video object segmentation, clusters of long-term point tracks have been used [3, 22, 23], assuming that points from the same object have similar tracks. In [19, 20, 25, 35], the appearance of potential object and background regions is modeled and combined with motion information. These methods produce results for individual videos and do not investigate relationships between videos and the objects they contain. Video object cosegmentation aims to segment a detailed mask of common object out of videos. This problem has been addressed with weak supervision such as an object label per video [33] and additional labels for a few frames that indicate whether the frames contain a target object or not [38].

Finally, spatio-temporal proposals of [16, 24] and action localization [39, 42] are relevant to our work as they also return spatio-temporal tubes as output. However, our method localizes an object through a single volume, whereas the proposals [16, 24] form a large number of hypotheses that have to be validated by post-processing. Furthermore, unlike action localization techniques [39, 42], our approach does not require any training data.

1.2. Proposed approach

We consider a set of videos v, each consisting of T frames (images) v_t ($t = 1, \ldots, T$), and denote by $R(v_t)$ a set of candidate regions identified in v_t by some separate bottom-up region proposal process [21]. We also associate with v_t a *matching neighborhood* $N(v_t)$ formed by the k closest frames w_u among all videos $w \neq v$, according to a robust criterion based on probabilistic Hough matching (see [4] and Section 2.1). The network structure defined by N links *frames* across *different* videos. We also link *regions* in successive frames of the *same* video, so that r_t in $R(v_t)$ and r_{t+1} in $R(v_{t+1})$ are *tracking neighbors* when there exists some point track originating in r_t and terminating in r_{t+1}. A *spatio-temporal tube* is any sequence $r = [r_1, \ldots, r_T]$ of temporal neighbors in the same video. Our goal is to find, for every video v in the input collection, the top tube r according to the criterion

$$\Omega_v(r) = \sum_{t=1}^{T} \varphi[r_t, v_t, N(v_t)] + \lambda \sum_{t=1}^{T-1} \psi(r_t, r_{t+1}), \quad (1)$$

where $\varphi[r_t, v_t, N(v_t)]$ is a measure of confidence for r_t being an object (foreground) region, given v_t and its matching neighbors, and $\psi(r_t, r_{t+1})$ is a measure of temporal consistency between r_t and r_{t+1}; λ is a weight on temporal consistency.

As will be shown in the sequel, given the matching network structure N, finding the top tube (or for that matter the top p tubes) for each video can be done efficiently using dynamic programming. The top tubes then help to find a better matching network structure since the tubes tend to focus on objects and disregard background clutters when finding matching neighborhoods. Thus we adopt an iterative process, alternating between steps where N is fixed and the top tubes are computed for each video, and steps where the top tubes are fixed, and used to update the matching network. After a few iterations, we stop, and finally pick the top scoring tube for each video. We dub this iterative process a *discovery and tracking* procedure since finding the tubes maximizing foreground confidence across videos is akin to unsupervised object discovery [4, 11, 12, 28, 31], whereas finding the tubes maximizing temporal consistency within a video is similar to object tracking [1, 2, 13, 26, 40, 41].

Interestingly, because we update the matching neighborhood structure at every iteration, our discovery and tracking procedure does much more than finding the spatio-temporal tubes associated with dominant objects: It also discovers the implicit neighborhood structure of frames associated with instances of the same class, which is a role normally left to supervisory information in the form of class labels in conventional image and video understanding methods. Indeed, as demonstrated by our experiments, our method can handle video collections featuring multiple object classes with minimal or zero supervision (it is, however, limited for the time being to one object instance per frame).

We describe in the next two sections the foreground confidence and temporal consistency terms in Eq. (1), before describing in Section 4 our discovery and tracking algorithm, presenting experiments in Section 5, and concluding in Section 6 with brief remarks about future work.

2. Foreground confidence

Our foreground confidence term is defined as a weighted sum of appearance- and motion-based confidences:

$$\varphi[r_t, v_t, N(v_t)] = \varphi_{\mathrm{a}}[r_t, v_t, N(v_t)] + \alpha\, \varphi_{\mathrm{m}}(r_t). \quad (2)$$

where α is a weight on motion-based confidence. For appearance-based confidence, we follow [4] and use a *standout score* based on region matching confidence. For motion-based confidence, we build on long-term point track clusters [3] and propose a *motion coherence score* that measures how well a box region aligns with motion clusters.

2.1. Appearance-based confidence

Given a set of images containing foreground objects of the same class with different backgrounds, object regions in an image are more likely to match with other images than background regions, and a region tightly enclosing the object stands out over its background. Recent work on unsupervised object discovery [4] implements this concept in a *standout score* based on a region matching algorithm, called probabilistic Hough matching (PHM). Here we extend the idea to video frames.

PHM is an efficient region matching algorithm which calculates scores for region matches using appearance and geometric consistency. Assume two sets of region proposals have been extracted from v_t and v_u: $R_t = R(v_t)$ and $R_u = R(v_u)$. Let $r_t = (f_t, l_t) \in R_t$ be a region with its 8×8 HOG descriptor f_t [7, 14] and its location l_t, *i.e.*, position and scale. The score for match $m = (r_t, r_u)$ is decomposed into an appearance term $m_{\mathrm{a}} = (f_t, f_u)$ and a geometry term $m_{\mathrm{g}} = (l_t, l_u)$. Let x denote the location offset of a potential object common to v_t and v_u. Given R_t and R_u, PHM evaluates the match score $c(m|R_t, R_u)$ by combining the Hough space vote $h(x|R_t, R_u)$ and the appearance similarity in a pseudo-probabilistic way:

$$\begin{aligned} c(m|R_t, R_u) &= p(m_{\mathrm{a}}) \sum_x p(m_{\mathrm{g}}|x) h(x|R_t, R_u), && (3)\\ h(x|R_t, R_u) &= \sum_m p(m_{\mathrm{a}}) p(m_{\mathrm{g}}|x), && (4) \end{aligned}$$

where $p(m_{\mathrm{a}})$ is the appearance-based similarity between two descriptors f_t and f_u, and $p(m_{\mathrm{g}}|x)$ is the likelihood of displacement $l_t - l_u$, which is defined as a Gaussian distribution centered on x. As noted in [4], this can be seen as a combination of bottom-up Hough space voting (Eq. [4]) and top-down confidence evaluation (Eq. [3]). Given neighbor frames $N(v_t)$ where an object in v_t may appear, the corresponding region saliency is defined as the sum of max-pooled match scores from R'_u to r:

$$g(r_t|R_t, R_u) = \sum_{v_u \in N(v_t)} \max_{r_u \in R_u} c\big((r_t, r_u)|R_t, R_u\big). \quad (5)$$

We omit the terms R_t and R_u in g for the sake of brevity afterwards. The region saliency $g(r_t)$ is high when r matches the neighbor frames well in terms of both appearance and geometric consistency. While useful as an evidence for foreground regions, the region saliency of Eq. (5) may be higher on a part than a whole object because part regions often match more consistently than entire object regions. To counteract this effect, a standout score measures how much the region r_t "stands out" from its potential backgrounds in terms of region saliency:

$$\begin{aligned} s(r_t) &= g(r_t) - \max_{r_{\mathrm{b}} \in B(r_t)} g(r_{\mathrm{b}}), \\ \text{s.t.}\quad B(r_t) &= \{r_{\mathrm{b}} | r_t \subsetneq r_{\mathrm{B}}, r_{\mathrm{b}} \in R_t\}, \quad (6) \end{aligned}$$

where $r_t \subsetneq r_{\mathrm{b}}$ indicates that region r_t is contained in region r_{b}. As can be seen from Eq.(5), the standout score $s(r_t)$ evaluates a foreground likelihood of r_t based on region matching between frame v_t and its neighbor frames from different videos $N(v_t)$. Now we denote it more explicitly using $s\big(r_t|v_t, N(v_t)\big)$. The appearance-based foreground confidence for region r_t is defined as the standout score of r_t:

$$\varphi_{\mathrm{a}}[r_t, v_t, N(v_t)] = s\big(r_t|v_t, N(v_t)\big). \quad (7)$$

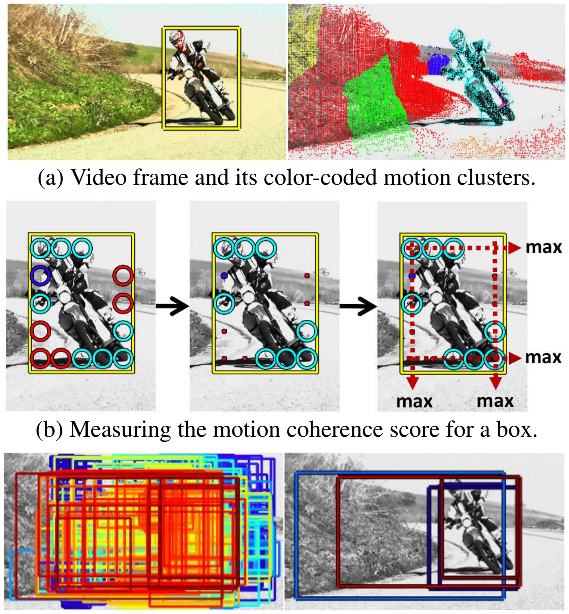
In practice, we rescale the score to the $[0, 1]$ in each frame.

2.2. Motion-based confidence

Motion is an important cue for localizing moving objects in videos [25] since these often exhibit motions that differentiate the objects from the background. To exploit this information, we build on long-term point tracks [3] and propose a *motion coherence score* for motion-based foreground confidence. These tracks are more "global" than conventional optical flow in the sense that they use more frames. Motion clusters based on the long-term tracks are even more "global" since they incorporate both temporal and spatial coherence. We propose to compute the *motion coherence score* for a box in three steps: (1) edge motion binning, (2) motion cluster weighting, (3) edge-wise max pooling. First, we divide the box into 5×5 cells, and construct bins along the four edges of the box as illustrated in Fig. 2. Then, for each bin b, we assign its cluster label l_b by majority voting using the tracks that falls into the bin. Bins with no tracks remain empty. Second, we compute a motion cluster weight for each cluster label i:

$$w(l) = \frac{\#\text{ of tracks of cluster } l \text{ within the box}}{\#\text{ of all tracks of cluster } l \text{ in the frame}}, \quad (8)$$

evaluating how much of the motion cluster the box includes, compared to the entire frame. The weight is assigned to the corresponding bin, and suppresses the effect of background clusters in the bins. Third, we select the bin with the maximum cluster weight along each edge, and define the sum



(a) Video frame and its color-coded motion clusters.

(b) Measuring the motion coherence score for a box.

(c) Heat map of the scores and the top 5 boxes.

Figure 2. (a) Given a video clip, its motion clusters are computed for each frame [3]. The example shows a frame (*left*) and its motion cluster with color coding (*right*). (b) Given a region box (yellow), the motion coherence score for the box is computed in three steps: box-boundary binning (*left*), cluster weighting (middle), and edge-wise max pooling (*right*). For the details, see text. (c) Heat map of the motion coherence scores (*left*) and the top 5 boxes with the best scores (*right*). (Best viewed in color.)

of the weights as the motion coherence score for the box, which is used for the motion-based confidence:

$$\varphi_{\mathrm{m}}(r_t) = \sum_{e\in\{\mathrm{L,R,T,B}\}} \max_{b\in E_e} w(l_b), \tag{9}$$

where e represents one of four edges of the box region (left, right, top, bottom), E_e a set of bins on the edge, and l_b the cluster label of bin b. This score is designed to be high for a box region that meets motion cluster boundaries (edge-wise max pooling) and contains the entire clusters (motion cluster weighting). Note that an object does not often fit a box shape correctly, but only touches the four edges. On this account, edge-wise max pooling provides a more robust score than average pooling on entire cells. This motion coherence score is useful to discover moving objects in video frames, and acts a complementary cue to the standout score in Section 2.1.

3. Temporal consistency

Regions with high foreground confidences may turn out to be temporally inconsistent. They can be misaligned due to imperfect confidence measures and ambiguous observations. Also, given multiple object instances of the same category, foreground regions may correspond to different instances in a video. Our temporal consistency term is used to handle these issues so that selected spatio-temporal tubes are temporally more stable and consistent. We exploit both appearance- and motion-based evidences for this purpose. We denote by $\psi_{\mathrm{a}}(r_t, r_{t+1})$ and $\psi_{\mathrm{m}}(r_t, r_{t+1})$ appearance- and motion-based terms, respectively. The consistency term of Eq. (1) is obtained as

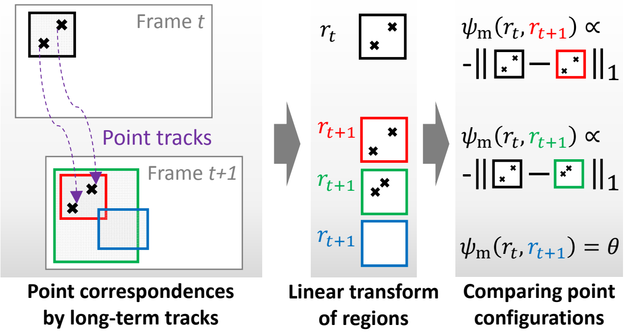

Frame t
Point tracks
Frame t+1
r_t
r_{t+1}
r_{t+1}
r_{t+1}
$\psi_{\mathrm{m}}(r_t, r_{t+1}) \propto$
$-\|\ \ -\ \ \|_1$
$\psi_{\mathrm{m}}(r_t, r_{t+1}) \propto$
$-\|\ \ -\ \ \|_1$
$\psi_{\mathrm{m}}(r_t, r_{t+1}) = \theta$
Point correspondences by long-term tracks
Linear transform of regions
Comparing point configurations

Figure 3. We compare two sets of corresponding points in consecutive regions by transforming them into a unit square from the regions. The configuration of points does not align with each other unless two regions match well (*e.g.*, *black* and *green*). The motion-based consistency uses the sum of distances between the corresponding points in the transformed domain. If two regions share no point track, we assign a constant θ as the consistency term. (Best viewed in color.)

$$\psi(r_t, r_{t+1}) = \psi_{\mathrm{a}}(r_t, r_{t+1}) + \psi_{\mathrm{m}}(r_t, r_{t+1}). \tag{10}$$

We describe these terms in the following subsections.

3.1. Appearance-based consistency

We use appearance similarity between two consecutive regions as a temporal consistency term. Region r_t is described by an 8×8 HOG descriptor f_t, as in Section 2.1, and the appearance-based consistency is defined as the opposite of the distance between descriptors:

$$\psi_{\mathrm{a}}(r_t, r_{t+1}) = -\|f_t - f_{t+1}\|_2, \tag{11}$$

which is rescaled in practice to cover $[0, 1]$ at each frame.

3.2. Motion-based consistency

Two consecutive regions r_t and r_{t+1} associated with the same object typically share the same point tracks, and configurations of the points in the two regions should be similar. Long-term point tracks [3] provide correspondences for such points across frames, which we exploit to measure the motion-based consistency between a pair of regions.

To compare the configurations of shared point tracks, we linearly transform each box region and internal point coordinates into a 1×1 unit square, as illustrated in Fig. 3. This normalization allows us to account for non-uniform scaling when comparing point configurations across different regions. Let p be an individual point track and p_t the position of p at frame t. Then, the position of p relative to region r_t is denoted by $\tau(p_t|r_t)$. If two consecutive regions r_t and r_{t+1} cover the same object and share a point track p, $\tau(p_t|r_t)$ and $\tau(p_{t+1}|r_{t+1})$ should be close to each other. The motion consistency $\psi_{\mathrm{m}}(r_t, r_{t+1})$ reflects this observation. Let P_{r_t} be the set of points occupied by region r_t. The motion-based consistency is defined as

$$\psi_{\mathrm{m}}(r_t, r_{t+1}) = -\sum_{p \in P_{r_t} \cap P_{r_{t+1}}} \frac{||\tau(p_t|r_t) - \tau(p_{t+1}|r_{t+1})||_1}{2 \mid P_{r_t} \cap P_{r_{t+1}} \mid}. \quad (12)$$

If r_t and r_{t+1} share no point track, we assign a constant value $\psi_{\mathrm{m}}(r_t, r_{t+1}) = \theta$, which is smaller than -1, the minimum value of $\psi_{\mathrm{m}}(r_t, r_{t+1})$, to penalize transitions between regions having no point correspondence. Through this consistency term, we can measure variations in spatial position, aspect ratio, and scale between regions at the same time.

4. Discovery and tracking algorithm

We initialize each tube r as an entire video (a sequence of entire frames), and alternate between (1) updating the neighborhood structure across videos and (2) optimizing $\Omega_v(r)$ within each video. The intuition is that better object discovery may lead to more accurate object tracking, and vice versa. These two steps are repeated for a few iterations until (near-) convergence. In our experiments, using more than 5 iterations does not improve performance. The number of neighbors for each frame is fixed as $k = 10$. The final result is obtained by selecting the best tube for each video after 5 iterations . As each video is independently processed at each iteration, the algorithm is easily parallelized.

Neighbor update. Given a localized tube r fixed for each video, we update the neighborhood structure N by k nearest-neighbor retrieval for each localized object region. At the first iteration, the nearest-neighbor search is based on distances between GIST descriptors [34] of frames as the tube r is initialized as the entire video. From the second iteration, the metric is defined as the appearance similarity between potential object regions localized at the previous iteration. Specifically, we select the top 20 region proposals inside potential object regions according to region saliency (Eq. [5]), and perform PHM between those small sets of regions. The similarity is then computed as the sum of all region saliency scores given by the matching. This selective region matching procedure allows us to perform efficient and effective retrieval for video frames.

Object relocalization. Given the neighborhood structure N, we optimize the objective of Eq. (1) for each video v. To exploit the tubes localized at the previous iteration, we confine region proposals in neighbor frames to those contained in the localized tube of the frames. This is done in Eq.(7) by substituting the neighbor frames of each frame v_t with the regions r_u localized in the frames: set $w_u = r_u$ for all w_u in $N(v_t)$. Before the optimization, we compute foreground confidence scores of region proposals, and select the top 100 among these according to their confidence scores. Only the selected regions are considered during optimization for efficiency. The objective of Eq.(1) is then efficiently optimized by dynamic programming (DP) [6, 26]. Note that using the p best tubes ($p = 5$ in all our experiments) for each video at each iteration (except the last one), instead of retaining only one candidate, increases the robustness of our approach. This agrees with the conclusions of [4] in the still image domain, and has also been confirmed empirically by our experiments. We obtain p best tubes by sequential DP, which iteratively removes the best tube and re-run dynamic programming again.

5. Implementation and results

Our method is evaluated on the YouTube-Object dataset [27], which consists of videos downloaded from YouTube by querying for 10 object classes from PASCAL VOC [10]. Each video of the dataset comes from a longer video and is segmented by automatic shot boundary detection. This dataset is challenging since the videos involve large camera motions, view-point changes, decoding artifacts, editing effects, and incorrect shot boundaries. Ground-truth boxes are given for a subset of the videos, and one frame is annotated per video. Following [17], our experiments are conducted on all the annotated videos.

We demonstrate the effectiveness of our method through various experiments. First, we evaluate our method in the conventional *colocalization* setting, where videos contain at least one object of a sample category. Our method is also tested in a fully unsupervised mode, where all videos from all classes of the dataset are mixed; we call this challenging setting *unsupervised object discovery*.

5.1. Implementation details

Key frame selection. We sample key frames from each video uniformly with stride 20, and our method is used only on the key frames. This is because temporally adjacent frames typically have redundant information, and it is time-consuming to process all the frames. Note that long-term point tracks enable us to utilize continuous motion information although our method works on temporally sparse key frames. To obtain temporally dense localization results, object regions in non-key frames are estimated by interpolating localized regions in temporally adjacent key-frames.

Parameter setting. The weight for the motion-based confidence α and that for the temporal consistency terms λ are set to 0.5 and 2, respectively. To penalize transitions between regions sharing no point track, θ is set to -2, which is smaller than the minimum value of ψ_{m} (see Section 3.2). The number of region candidates for object relocalization

Table 1. Colocalization performance in CorLoc on the YouTube-Object dataset.

Method	aeroplane	bird	boat	car	cat	cow	dog	horse	motorbike	train	Average
Prest *et al.* [27]	51.7	17.5	34.4	34.7	22.3	17.9	13.5	26.7	41.2	25.0	28.5
Joulin *et al.* [17]	25.1	31.2	27.8	38.5	41.2	28.4	33.9	35.6	23.1	25.0	31.0
F(A)	38.2	67.3	30.4	75.0	28.6	65.4	38.3	46.9	52.0	25.9	46.8
F(A)+T(M)	44.4	68.3	31.2	76.8	30.8	70.9	56.0	55.5	58.0	27.6	51.9
F(A)+T(A,M)	52.9	72.1	55.8	79.5	30.1	67.7	56.0	57.0	57.0	25.0	55.3
F(A,M)+T(A,M)	56.5	66.4	58.0	76.8	39.9	69.3	50.4	56.3	53.0	31.0	55.7

Table 2. Unsupervised object discovery performance in CorLoc on the YouTube-Object dataset.

Method	aeroplane	bird	boat	car	cat	cow	dog	horse	motorbike	train	Average
Brox and Malik [3]	53.9	19.6	38.2	37.8	32.2	21.8	27.0	34.7	45.4	37.5	34.8
Papazoglou and Ferrari [25]	65.4	67.3	38.9	65.2	46.3	40.2	65.3	48.4	39.0	25.0	50.1
F(A,M)+T(A,M)	55.2	58.7	53.6	72.3	33.1	58.3	52.5	50.8	45.0	19.8	49.9

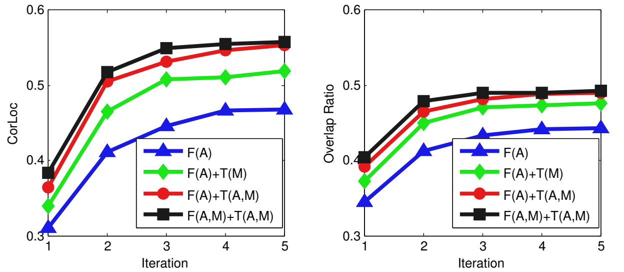


Figure 4. Average CorLoc scores (*left*) and average overlap ratios (*right*) versus iterations on the YouTube-Object dataset in the colocalization setting.

(Section 4) is restricted to 100 to reduce computation. The other parameters $k = 10$ and $p = 5$ in Section 4 are adopted directly from [4]. All the parameters are fixed for all experiments. For analysis and discussion of the parameters, see Section 5.5.

Execution time. Our method is implemented in MATLAB without sophisticated optimization. On a machine with a Xeon CPU (2.6GHz, 12 cores), it currently takes about 60 hours to handle the entire dataset with 5 iterations.

5.2. Evaluation metrics

Our method not only discovers and localizes objects, but also reveals the topology between different videos and the objects they contain. We evaluate our results on those two tasks with different measures.

Localization accuracy is measured using CorLoc [17, 25, 27], which is defined as the percentage of images correctly localized according to the PASCAL criterion: $\frac{area(r_p \cap r_{gt})}{area(r_p \cup r_{gt})} > 0.5$, where r_p is the predicted region and r_{gt} is the ground-truth.

In the unsupervised object discovery setting, we measure the quality of the topology revealed by our method as well as localization performance. To this end, we first employ the CorRet metric, originally introduced in [4], which is defined in our case as the mean percentage of retrieved nearest neighbor frames that belongs to the same class as the target video. We also measure the accuracy of nearest neighbor classification, where a query video is classified by the most frequent labels of its neighbor frames retrieved by our method. The classification accuracy is reported by the top-k error rate, which is the percentage of videos whose ground-truth labels do not belong to the k most frequent labels of their neighbor frames. All the evaluation metrics are given as percentages.

5.3. Object colocalization per class

We compare our method with two colocalization methods for videos [17, 27]. We also compare our method with several of its variants to highlight benefits of each of its components. Specifically, the components of our method are denoted by combinations of four characters: ‘F’ for foreground confidence, ‘T’ for temporal consistency, ‘A’ for appearance, and ‘M’ for motion. For example, F(A) means foreground saliency based only on appearance (*i.e.*, φ_a), and T(A,M) indicates temporal smoothness based on both of appearance and motion (*i.e.*, $\psi_a + \psi_m = \psi$). Our full model corresponds to F(A,M)+T(A,M).

Quantitative results are summarized in Table 1. Our method outperforms the previous state of the art in [17] on the same dataset, with a substantial margin. Comparing our full method to its simpler versions, we observe that performance improves by adding each of the temporal consistency terms. The motion-based confidence can damage performance when motion clusters include only a part of object (*e.g.*, “bird”, “dog”) and/or background has distinctive clusters due to complex 3D structures (*e.g.*, car, motorbike). However, it enhances localization when the object is highly non-rigid (*e.g.*, “cat”) and/or is clearly separated from the background by motion (*e.g.*, “aeroplane”, “boat”). In the “train” class case, where our method without motion-based confidence often localizes only a part of long trains, the motion-based confidence improves localization accuracy. Figure 4 shows the performance of our method over iterations. Our full method performs better than its variants at every iteration, and most quickly improves both of CorLoc score and overlap ratio in early stages.

Qualitative examples are shown in Fig. 5 and 6, where the regions localized by our full model are compared with those of F(A), which relies only on image-based informa-

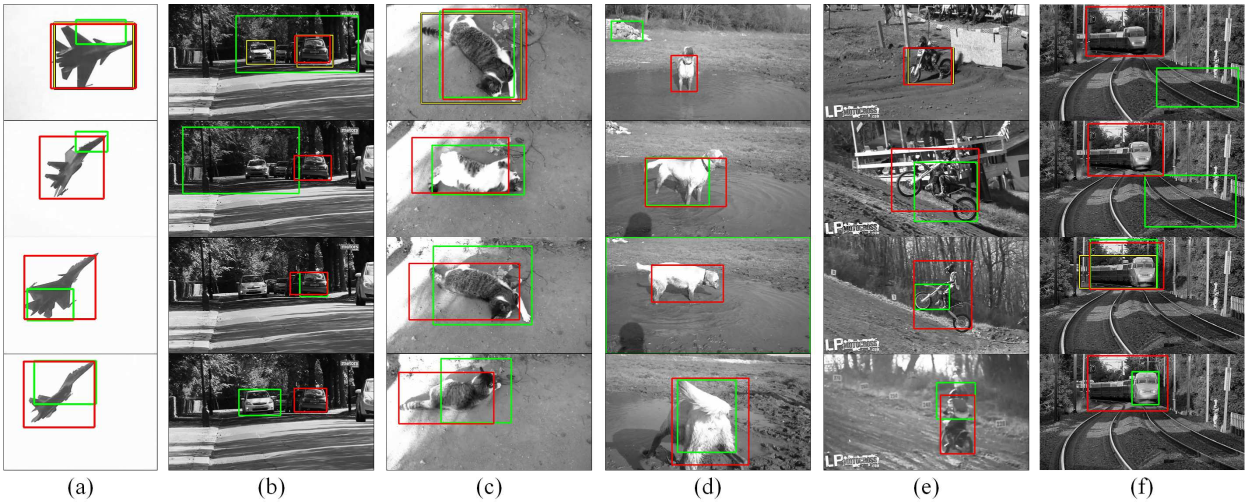


Figure 5. Visualization of examples that are correctly localized by our full method: (*red*) our full method, (*green*) our method without motion information, (*yellow*) ground-truth localization. The sequences come from (a) "aeroplane", (b) "car", (c) "cat", (d) "dog", (e) "motorbike", and (f) "train" classes. Frames are ordered by time from top to bottom. The localization results of our full method are spatio-temporally consistent. On the other hand, the simpler version often fails due to pose variations of objects (a, c–f) or produces inconsistent tracks when multiple target objects exist (b). More results are included in the supplementary file. (Best viewed in color.)

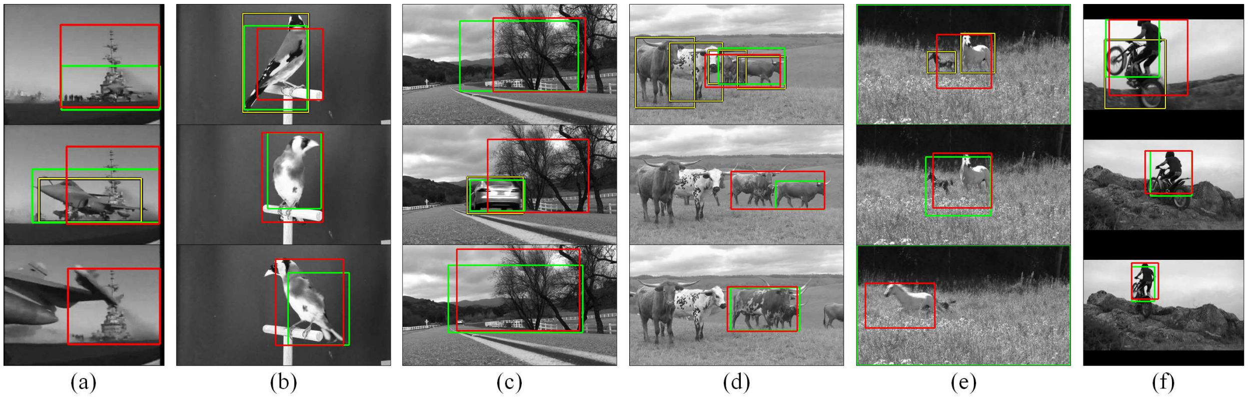


Figure 6. Examples incorrectly localized by our full method: (*red*) our full method, (*green*) our method without motion information, (*yellow*) ground-truth localization. The sequences come from (a) "aeroplane", (b) "bird", (c) "car", (d) "cow", (e) "horse", and (f) "motorbike". Frames are ordered by time from top to bottom. Our full method fails when background looks like an object and is spatio-temporally more consistent than the object (a, c), or the boundaries of motion clusters include the multiple objects or background together (b, d–f). The localization results in (b) and (f) are reasonable although they are incorrect according to the PASCAL criterion. (Best viewed in color.)

Table 3. CorRet scores and top-k error rates of our method on the YouTube-Object dataset in the fully unsupervised setting.

Metric	aeroplane	bird	boat	car	cat	cow	dog	horse	motorbike	train	Average
CorRet	66.9	36.1	49.5	51.8	15.9	30.6	20.7	22.6	15.3	45.5	35.5
Top-1 error rate	12.1	51.9	34.1	25.0	84.2	45.7	70.2	73.4	83.0	33.6	51.3
Top-2 error rate	4.6	46.2	10.9	18.8	60.9	24.4	41.1	49.2	63.0	20.7	34.0

tion. F(A) already outperforms the previous state of the art, but its results are often temporally inconsistent when the object undergoes severe pose variation or multiple target objects exist in a video. We handle this problem by enforcing temporal consistency on the solution.

5.4. Unsupervised object discovery

In the unsupervised setting, where videos with different object classes are all mixed together, our method still outperforms existing video colocalization techniques even though it does not use any supervisory information, as summarized in Table 2. It performs slightly worse than the state of the art in video segmentation [25], which uses a foreground/background appearance model. Note however that (1) such a video-specific appearance model would probably further improve our localization accuracy; and (2) our method attacks a more difficult problem, and, unlike [25], discovers the underlying topology of the video collection.

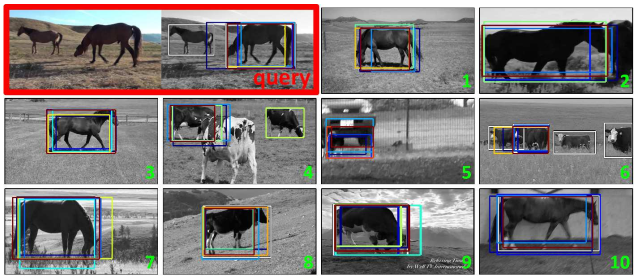
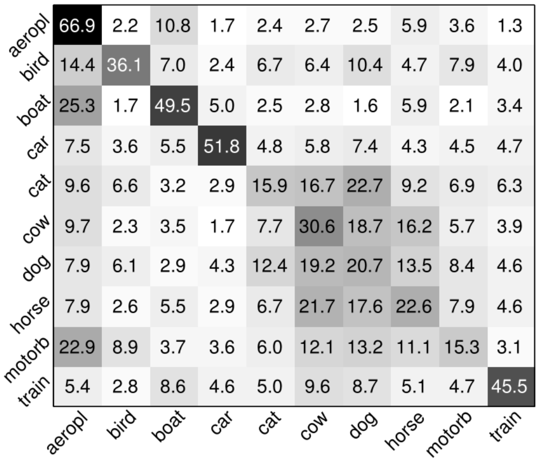


Figure 7. A query frame (bold outer box) from the "horse" class and its nearest neighbor frames at the last iteration of the unsupervised object discovery and tracking. The top-5 object candidates (inner boxes) of the nearest neighbors look similar with those of the query, although half of them come from the "cow" class (4th, 6th, 8th, and 9th) or the "car" class (5th).



	aeropl	bird	boat	car	cat	cow	dog	horse	motorb	train
aeropl	66.9	2.2	10.8	1.7	2.4	2.7	2.5	5.9	3.6	1.3
bird	14.4	36.1	7.0	2.4	6.7	6.4	10.4	4.7	7.9	4.0
boat	25.3	1.7	49.5	5.0	2.5	2.8	1.6	5.9	2.1	3.4
car	7.5	3.6	5.5	51.8	4.8	5.8	7.4	4.3	4.5	4.7
cat	9.6	6.6	3.2	2.9	15.9	16.7	22.7	9.2	6.9	6.3
cow	9.7	2.3	3.5	1.7	7.7	30.6	18.7	16.2	5.7	3.9
dog	7.9	6.1	2.9	4.3	12.4	19.2	20.7	13.5	8.4	4.6
horse	7.9	2.6	5.5	2.9	6.7	21.7	17.6	22.6	7.9	4.6
motorb	22.9	8.9	3.7	3.6	6.0	12.1	13.2	11.1	15.3	3.1
train	5.4	2.8	8.6	4.6	5.0	9.6	8.7	5.1	4.7	45.5

Figure 8. Confusion matrix of nearest neighbor retrieval. Rows correspond to query classes and columns indicate retrieved classes. Diagonal elements correspond to the CorRet values on Table 3.

The quality of nearest-neighbor retrieval is measured by CorRet and quantified in Table 3. Even in the case where some neighbors do not come from the same class as the query, object candidates in the neighbor frames usually resemble to those in the query frame, as illustrated in Fig. 7. To illustrate the recovered topology between classes, we provide a confusion matrix of the retrieval results in Fig. 8, showing that most classes are most strongly connected to themselves, and some classes with somewhat similar appearances (*e.g.*, "cat", "dog", "cow", and "horse") have some connections between them. Finally, we measure the accuracy of nearest neighbor classification that is based on neighbor frames provided by our method and their ground-truth labels. The classification accuracy in top-1 and top-2 error rates is summarized in Table 3. The error rates are low when the query class usually shows unique appearances (*e.g.*, "aeroplane", "boat", "car", and "train"), and high if there are other classes with similar appearances (*e.g.*, "cat", "dog", "cow", and "horse").

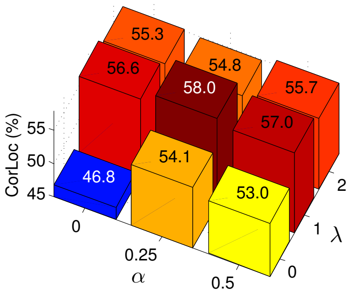


Figure 9. Average CorLoc scores for different values of the two weight parameters α and λ on the YouTube-Object dataset in the colocalization setting. The CorLoc score of our full method (55.7) in Table 1 corresponds to $\alpha = 0.5, \lambda = 2$.

5.5. Effect of parameters α and λ

To study the influence of weight parameters α and λ, we have conducted additional experiments by evaluating our method on a 3×3 grid of weight parameters: $\alpha \in \{0, 0.25, 0.5\}$ and $\lambda \in \{0, 1, 2\}$. The results in the colocalization setting are shown in Fig. 9. Note that F(A) corresponds to $\alpha = 0$, $\lambda = 0$. A substantial improvement over F(A) is achieved by assigning a non-zero value to α or λ in all cases, which shows that both motion-based confidence and temporal consistency contribute to the performance. Also, when both of α and λ are non-zero, the score varies between 53.0 and 58.0, which is relatively stable. We believe that similar results will be observed in the totally unsupervised setting, although we did not investigate the effect of the parameters in that setting.

The best performance in this experiment has been acheived with $\alpha = 0.25$, $\lambda = 1$, which outperforms the results reported in Table 1. Note that in the previous experiments, we did not optimize α and λ for the entire dataset, but selected their values from only a few candidates validated in a small portion of the data. In a totally unsupervised setting such as ours, there is no perfect way to optimize those parameters.

6. Discussion and Conclusion

We have proposed a novel approach to localizing objects in an unlabeled video collection by a combination of object discovery and tracking. We have demonstrated the effectiveness of the proposed method on the YouTube-Object dataset, where it significantly outperforms the state of the art in colocalization even though it uses much less supervision. Some issues still remain for further exploration. As it stands, our method is not appropriate for videos with a single dominant background and highly non-rigid object (*e.g.*, the UCF-sports dataset). Next on our agenda is to address these issues, using for example video stabilization and foreground/background models [19, 20, 25].

Acknowledgments. This work was supported by the ERC grants Activia, Allegro, and VideoWorld; and by the Institut Universitaire de France and a Google research award.

References

[1] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. In *CVPR*, 2011. 2

[2] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. V. Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *PAMI*, 2011. 2

[3] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, pages 282–295, 2010. 2, 3, 4, 6

[4] M. Cho, S. Kwak, C. Schmid, and J. Ponce. Unsupervised object discovery and localization in the wild: Part-based matching using bottom-up region proposals. In *CVPR*, 2015. 1, 2, 3, 5, 6

[5] R. G. Cinbis, J. Verbeek, and C. Schmid. Multi-fold MIL Training for Weakly Supervised Object Localization. In *CVPR*, 2014. 1

[6] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2001. 5

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 3

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 1

[9] T. Deselaers, B. Alexe, and V. Ferrari. Localizing Objects While Learning Their Appearance. In *ECCV*, 2010. 1

[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. 1, 5

[11] A. Faktor and M. Irani. “Clustering by composition”–Unsupervised discovery of image categories. In *ECCV*, 2012. 2

[12] K. Grauman and T. Darrell. Unsupervised learning of categories from sets of partially matching image features. In *CVPR*, 2006. 2

[13] S. Hare, A. Saffari, and P. H. S. Torr. Struck: Structured output tracking with kernels. In *ICCV*, 2011. 2

[14] B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. In *ECCV*, 2012. 3

[15] S. Hong, S. Kwak, and B. Han. Orderless tracking through model-averaged posterior estimation. In *ICCV*, 2013. 2

[16] M. Jain, J. van Gemert, H. Jegou, P. Bouthemy, and C. Snoek. Action localization with tubelets from motion. In *CVPR*, pages 740–747, June 2014. 2

[17] A. Joulin, K. Tang, and L. Fei-Fei. Efficient Image and Video Co-localization with Frank-Wolfe Algorithm. In *ECCV*, 2014. 1, 2, 5, 6

[18] G. Kim and A. Torralba. Unsupervised detection of regions of interest using iterative link analysis. In *NIPS*, 2009. 2

[19] S. Kwak, T. Lim, W. Nam, B. Han, and J. H. Han. Generalized background subtraction based on hybrid inference by belief propagation and bayesian filtering. In *ICCV*, 2011. 2, 8

[20] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *CVPR*, 2011. 2, 8

[21] S. Manen, M. Guillaumin, and L. Gool. Prime object proposals with randomized Prim’s algorithm. In *ICCV*, 2013. 2

[22] P. Ochs and T. Brox. Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions. In *ICCV*, 2011. 2

[23] P. Ochs and T. Brox. Higher order motion models and spectral clustering. In *CVPR*, 2012. 2

[24] D. Oneata, J. Revaud, J. Verbeek, and C. Schmid. Spatio-temporal object detection proposals. In *ECCV*, pages 737–752. 2014. 2

[25] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, pages 1777–1784, Dec 2013. 2, 3, 6, 7, 8

[26] H. Pirsiavash, D. Ramanan, and C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, pages 1201–1208, June 2011. 1, 2, 5

[27] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, pages 3282–3289, 2012. 1, 2, 5, 6

[28] B. Russell, W. Freeman, A. Efros, J. Sivic, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006. 2

[29] G. Sharir and T. Tuytelaars. Video object proposals. *CVPRW*, 2012. 1

[30] Z. Shi, T. M. Hospedales, and T. Xiang. Bayesian joint topic modelling for weakly supervised object localisation. In *ICCV*, 2013. 1

[31] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering object categories in image collections. In *ICCV*, 2005. 2

[32] K. Tang, A. Joulin, L.-j. Li, and L. Fei-Fei. Co-localization in Real-World Images. In *CVPR*, 2014. 1, 2

[33] K. Tang, R. Sukthankar, J. Yagnik, and L. Fei-Fei. Discriminative segment annotation in weakly labeled video. In *CVPR*, 2013. 2

[34] A. Torralba, R. Fergus, and Y. Weiss. Small codes and large image databases for recognition. In *CVPR*, 2008. 5

[35] R. Trichet and R. Nevatia. Video segmentation with spatio-temporal tubes. In *AVSS*, pages 330–335, Aug 2013. 2

[36] F. Wang, Q. Huang, and L. Guibas. Image co-segmentation via consistent functional maps. In *ICCV*, 2013. 2

[37] F. Wang, Q. Huang, M. Ovsjanikov, and L. J. Guibas. Unsupervised multi-class joint image segmentation. In *CVPR*, 2014. 2

[38] L. Wang, G. Hua, R. Sukthankar, J. Xue, and N. Zheng. Video object discovery and co-segmentation with extremely weak supervision. In *CVPR*, 2014. 1, 2

[39] T. Wang, S. Wang, and X. Ding. Detecting human action as the spatio-temporal tube of maximum mutual information. *TCSVT*, 24(2):277–290, Feb 2014. 2

[40] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *CVPR*, 2013. 2

[41] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 2006. 2

[42] G. Yu, J. Yuan, and Z. Liu. Propagative Hough voting for human activity recognition. In *ECCV*, pages 693–706. 2014. 2