Common Action Discovery and Localization in Unconstrained Videos

Jiong Yang[1] Junsong Yuan[2]
[1]Interdisciplinary Graduate School [2]School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore
yang0374@e.ntu.edu.sg, jsyuan@ntu.edu.sg

Abstract

Similar to common object discovery in images or videos, it is of great interests to discover and locate common actions in videos, which can benefit many video analytics applications such as video summarization, search, and understanding. In this work, we tackle the problem of common action discovery and localization in unconstrained videos, where we do not assume to know the types, numbers or locations of the common actions in the videos. Furthermore, each video can contain zero, one or several common action instances. To perform automatic discovery and localization in such challenging scenarios, we first generate action proposals using human prior. By building an affinity graph among all action proposals, we formulate the common action discovery as a subgraph density maximization problem to select the proposals containing common actions. To avoid enumerating in the exponentially large solution space, we propose an efficient polynomial time optimization algorithm. It solves the problem up to a user specified error bound with respect to the global optimal solution. The experimental results on several datasets show that even without any prior knowledge of common actions, our method can robustly locate the common actions in a collection of videos.

1. Introduction

Given a collection of unlabeled videos as shown in Figure 1, can we automatically discover and locate the common actions that frequently appear in these videos? It is worth noting that the video collection may contain multiple types of common actions which are not known in advance, and each video can contain zero, one or several common action instances. Similar to common object discovery in images [22, 33, 46] or videos [18], finding common actions can benefit many video analytics applications such as video summarization [23], search [43, 44] and labeling.

However, compared with previous success of common object discovery in images and videos [16, 18, 21, 35], common action discovery is much less explored due to the following challenges. First, as we do not know in advance the types or locations of the actions that are common in the given dataset, we have to perform the discovery and localization simultaneously. Given a collection of unlabeled videos, we need to automatically identify a set of spatio-temporal bounding boxes that capture the common actions. Second, similar actions may also appear differently due to view point variation, scale variation or camera motion. It is not a trivial task to automatically associate these common actions. Last but not the least, besides common actions, videos may also contain dynamic backgrounds or uncommon actions, it is thus critical to differentiate such "noisy motions" from common actions.

Figure 1. Assuming that we are given a set of unlabeled videos (each frame represents a video), we would like to automatically discover and locate the common human actions in these videos. The common actions to be discovered and located are denoted by bounding boxes. Some videos contain one or multiple common actions, while some videos contain no common actions.

To address the above challenges of common action discovery and localization, we first use human prior, *i.e.*, human detector, to generate spatio-temporal action proposals in each video. However, it is inevitable that some proposals may contain dynamic background, uncommon actions or only partially capture the common actions. In order to stand out the proposals containing common actions from the initial proposal corpus, we build an affinity graph of the action proposals, and formulate the common action discov-

ery as a subgraph density maximization problem. Instead of using the average degree subgraph density [13] in which the average regularization is usually too strict for our co-localization problem, we propose a different subgraph density measure that relaxes the average regularization to recall more common actions. To avoid enumerating in the exponentially large solution space, we propose an efficient polynomial time algorithm to effectively find the optimal subgraph that captures common actions. The proposed algorithm solves the proposed formulation within a user specified error bound with respect to the global optimal solution. A tighter bound requires more computation.

The experimental results on several datasets show that even without any prior knowledge of common actions, the proposed method can robustly locate common actions in unconstrained videos, where each video can contain zero, one or several common actions. The extensive comparisons with other graph-based pattern discovery methods, *e.g.*, [2, 13, 27, 48, 49], as well as one recent video object co-localization method [16] validate the effectiveness of our method in the problem of common action co-localization.

2. Related Work

Our work is related to the video object co-localization methods, weakly supervised action co-localization methods and the maximum average degree density subgraph selection methods. We briefly discuss them in this section.

Video Object Co-localization Video object co-localization aims to locate the common objects in a video collection. The method in [33] performs image object co-localization by labeling image object proposals through quadratic programming. [18] extends [33] to perform video object co-localization by enforcing temporal consistency in the labeling process. The method in [4] performs image object co-localization by assigning a commonness score to each image object proposal using part based probabilistic Hough voting. [21] extends [4] to perform video object co-localization by combining the proposals' motion evidence scores with [21]'s commonness scores. [16] performs video object co-localization by assigning a co-saliency score to each image object proposal tubelet. [35] first generates object tracklets and then performs semantic video object co-segmentation by tracklet co-selection. [47] also generates object tracklets but performs the co-segmentation by finding the maximum weighted clique in a completely connected tracklet graph. The methods in [11, 12, 39, 40] co-segment the common object by energy minimization optimization in a spatio-temporal proposal [9] or superpixel [1] graph. However, most of the above video co-localization methods are for object instead of action co-localization. Moreover, they mostly assume each video contains exactly one common object and rarely explore the fully unconstrained scenario like us.

Action Co-localization There are also several weakly supervised action detection and localization methods [5, 6, 7, 8, 14, 24, 25, 31, 41], but they require video level labels to perform co-localization. [25] proposes a matrix completion approach to the problem of weakly supervised learning for multi-label learning. The methods in [24], [31] and [41] first extract action proposals, and then apply multiple instance learning to locate the labeled action. The method in [24] also requires point annotations to perform localization. The methods in [5, 14] perform action co-localization only in pairs of videos. [6, 7, 8] focus on the discovery of repeated articulated local motion patterns of the same object among the given videos. [45] operates in a less constrained scenario but it only provides temporal localizations.

Maximum Density Subgraph Selection Our work is also related to the subgraph selection method proposed in [13], which selects the subgraph with the maximum average degree density. However, this formulation is not suitable to our problem due to the strict average regularization. We relax this regularization in this work and propose an efficient optimization approach for this new formulation.

3. Proposed Method

In this section, we introduce the proposed action co-localization framework. The input is a collection of unlabeled videos, and the output is spatio-temporal localizations of the common actions appearing frequently in the videos. The proposed method is comprised of two steps. The first step is to extract action proposals from the input video collection. Each action proposal is a spatio-temporal tube, *i.e.*, a temporal sequence of bounding boxes, that locates an action instance. However, besides capturing the common actions, the proposals may also contain noisy background or actions that are not common in the dataset. Hence, the second step is to select the action proposals containing common actions from the initial proposal corpus.

In order to stand out the proposals containing common actions, we utilize the confidence score of each proposal as well as the similarities among the proposals. The former helps to reject the proposals containing non-action background and the latter helps to identify proposals containing common actions from those containing non-common actions. To integrate these two cues in a unified framework, we formulate it as a node selection problem in a graph $\mathbb{G} = (\mathcal{T}, \mathcal{E})$, where $\mathcal{T}$ denotes the collection of nodes, *i.e.*, action proposals, and $\mathcal{E}$ denotes the collection of edges. Node weights represent the quality scores of the proposals, and edge weights represent the semantic similarities between action proposals. Intuitively, the purpose is to select a subgraph in which most of the nodes have high quality scores and are densely connected to each other. Let t_i^j denote the i^{th} proposal in the j^{th} video, s_i^j denote the node

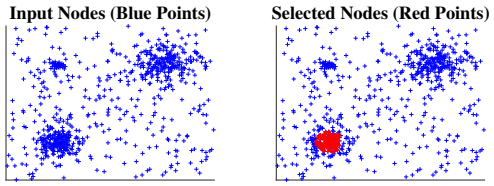

Figure 2. An illustration on the selection results of the classic average degree density maximization formulation defined by Eq. (1) and (2). Node weights are set to zero for simplicity.

weight of t_i^j, and $w(t_i^j, t_p^q)$ denote the edge weight between node t_i^j and t_p^q. A classic formulation to make such a selection is the average degree density maximization formulation in [13]:

$$\mathcal{A}^* = \arg\max_{\mathcal{A}\subseteq\mathcal{T}} D(\mathcal{A}), \tag{1}$$

where $\mathcal{A}$ is the subgraph containing the selected nodes and $D(\mathcal{A})$ is the average degree subgraph density defined as

$$D(\mathcal{A}) = \frac{\sum_{t_i^j\in\mathcal{A}} s_i^j + \sum_{\{t_i^j,t_p^q\}\subseteq\mathcal{A}} w(t_i^j, t_p^q)}{|\mathcal{A}|}, \tag{2}$$

where $|\cdot|$ is the cardinality of a set. The nominator computes the total node and edge weights of the selected subgraph $\mathcal{A}$, and the denominator regularizes the subgraph size. Although it can be solved efficiently using the method in [13], it is not suitable for our problem because the pure average regularization in the denominator is too strong. The selection of a larger subgraph will significantly decrease the subgraph density. Thus, it always favors very small subgraphs and leads to low recall. An example is shown in Figure 2. The selection completely misses the top two modes as well as the outer region of the bottom mode. Hence, to overcome this problem, we propose to use a relaxed regularization in the subgraph density definition:

$$D(\mathcal{A}) = \frac{\lambda\times\sum_{t_i^j\in\mathcal{A}} s_i^j + \sum_{\{t_i^j,t_p^q\}\subseteq\mathcal{A}} w(t_i^j, t_p^q)}{\eta + |\mathcal{A}|}. \tag{3}$$

Here λ is a parameter balancing the node and edge weights, and η weakens the subgraph size regularization which allows the selection of more nodes to improve recall. We name this new density definition as η-density. When η is zero, it is equivalent to the classic average degree density and only a small number of nodes will be selected. When η approaches to infinity, the regularization vanishes and all the nodes will be selected. However, the addition of η invalidates the optimization algorithm in [13], and enumerating all possible solutions is not feasible as the number of possible subgraphs is exponential to the problem size.

In this paper, we propose a polynomial time algorithm to solve Eq. (1) and (3). The proposed method is applicable to any η values and solves the problem within a user specified error bound with respect to the global optimal solution, although a tighter bound requires more computation.

In the following, we first present the construction of the affinity graph in Section 3.1, and then introduce the proposed optimization algorithm to Eq. (1) and (3) in Section 3.2. The details of action proposal generation and description are presented in Section 3.3.

3.1. Affinity Graph Construction

Given all the action proposals $\{t_i^j\}$ and their feature descriptions $\{f_i^j\}$, we build an ϵ-neighborhood [38] affinity graph, $\mathbb{G} = (\mathcal{T}, \mathcal{E})$, using all the proposal nodes. Node t_i^j and t_p^q will be linked only if $\|f_i^j - f_p^q\|_2 \leq \epsilon$, where $\|\cdot\|_2$ denotes the ℓ_2 distance and ϵ is the bandwidth for graph construction. The edge weight $w(t_i^j, t_p^q)$ is computed as

$$w(t_i^j, t_p^q) = \exp(\frac{-\|f_i^j - f_p^q\|_2^2}{2\times\beta^2}), \tag{4}$$

where β is computed as

$$\beta = \frac{\sum_{(t_i^j,t_p^q)\in\mathcal{E}} \|f_i^j - f_p^q\|_2}{|\mathcal{E}|}, \tag{5}$$

and $|\mathcal{E}|$ is the number of edges in the graph.

3.2. Density Maximization Optimization

When $\eta = 0$, Eq. (1) and (3) are the classic average degree density maximization formulation which can be solved efficiently by the method in [13]. However, the addition of a non-zero parameter η in Eq. (3) invalidates the original approach in [13], and enumerating in the exponentially large solution space is computationally infeasible. Hence, in this section, we propose a polynomial time algorithm that generalizes the approach in [13] to any non-zero η values. The proposed method uses a binary search strategy to find the optimal density $D(\mathcal{A}^*)$, *i.e.*, given the current lower bound l and upper bound u on $D(\mathcal{A}^*)$, we first check if our new guess $g = \frac{u+l}{2}$ defines a lower or upper bound on $D(\mathcal{A}^*)$ and then shrink the search space by half. A candidate subgraph whose η-density falls within the current bound is maintained during bound update. In the following, we introduce how to perform this bound check in the binary search process. We ignore λ in Eq. (3) without loss of generality.

Bound Check: To perform the bound check on our current guess g on $D(\mathcal{A}^*)$, we add two auxiliary nodes, *i.e.*, source node s and sink node t, to the original affinity graph. Both s and t are connected to all the original nodes as shown in Figure 3. Let d_i^j denote the degree of node t_i^j, *i.e.*, the weight summation of all the edges connected to t_i^j, the newly added source to node weight $\omega(s, t_i^j)$ and node to sink weight $\omega(t_i^j, t)$ are defined as

$$w(s, t_i^j) = m, \tag{6}$$

$$w(t_i^j, t) = m + 2\times g - d_i^j - 2\times s_i^j, \tag{7}$$

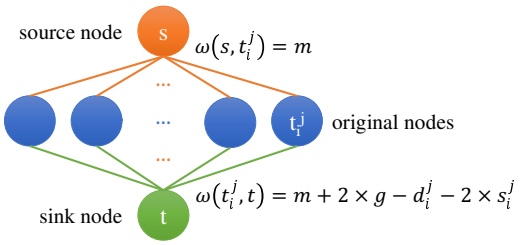

Figure 3. An illustration of the affinity graph with the addition of the source and sink nodes. The edges between the original nodes are omitted for simplicity.

where the variable m is defined as

$$m = \max_{t_i^j \in \mathcal{T}} (d_i^j + 2 \times s_i^j). \tag{8}$$

Note that all the newly added edge weights are non-negative based on the definition. Now let's cut the new graph by dividing the nodes into two disjoint subgraphs in which one subgraph contains the source node and another subgraph contains the sink node. Given an arbitrary cut, we denote the subgraph containing the source and sink node as $\mathcal{A}_s$ and $\mathcal{A}_t$, respectively. The cut capacity $c(\mathcal{A}_s, \mathcal{A}_t)$ is defined as the summation of the edge weights along the cut boundary. Interestingly, $c(\mathcal{A}_s, \mathcal{A}_t)$ is related to the η-density of subgraph $\mathcal{A}_s$, *i.e.*, $D(\mathcal{A}_s)$, as follows:

$$\begin{aligned}
& c(\mathcal{A}_s, \mathcal{A}_t) \\
&= \sum_{t_i^j \in \mathcal{A}_t} w(s, t_i^j) + \sum_{t_i^j \in \mathcal{A}_s} w(t_i^j, t) + \sum_{t_i^j \in \mathcal{A}_s, t_p^q \in \mathcal{A}_t} w(t_i^j, t_p^q) \\
&= m \times |\mathcal{A}_t| + \Big(m \times |\mathcal{A}_s| + 2 \times g \times |\mathcal{A}_s| \\
&\quad - \sum_{t_i^j \in \mathcal{A}_s} d_i^j - 2 \times \sum_{t_i^j \in \mathcal{A}_s} s_i^j \Big) + \sum_{t_i^j \in \mathcal{A}_s, t_p^q \in \mathcal{A}_t} w(t_i^j, t_p^q) \\
&= m \times |\mathcal{T}| + 2 \times g \times |\mathcal{A}_s| - 2 \times g \times \eta + 2 \times g \times \eta \\
&\quad - \sum_{t_i^j \in \mathcal{A}_s} d_i^j - 2 \times \sum_{t_i^j \in \mathcal{A}_s} s_i^j + \sum_{t_i^j \in \mathcal{A}_s, t_p^q \in \mathcal{A}_t} w(t_i^j, t_p^q) \\
&= m \times |\mathcal{T}| - 2 \times g \times \eta + 2 \times (|\mathcal{A}_s| + \eta) \times \Big(g \\
&\quad - \frac{\sum_{t_i^j \in \mathcal{A}_s} s_i^j}{|\mathcal{A}_s| + \eta} - \frac{\sum_{t_i^j \in \mathcal{A}_s} d_i^j - \sum_{t_i^j \in \mathcal{A}_s, t_p^q \in \mathcal{A}_t} w(t_i^j, t_p^q)}{2 \times (|\mathcal{A}_s| + \eta)} \Big) \\
&= m \times |\mathcal{T}| - 2 \times g \times \eta \\
&\quad + 2 \times (|\mathcal{A}_s| + \eta) \times (g - D(\mathcal{A}_s)).
\end{aligned} \tag{9}$$

Let $c^* = c(\mathcal{A}_s^*, \mathcal{A}_t^*)$ denote the minimum cut capacity on the current graph:

$$(\mathcal{A}_s^*, \mathcal{A}_t^*) = \arg \min_{\mathcal{A}_s \cap \mathcal{A}_t = \emptyset, \mathcal{A}_s \cup \mathcal{A}_t = \mathcal{T}, s \in \mathcal{A}_s, t \in \mathcal{A}_t} c(\mathcal{A}_s, \mathcal{A}_t). \tag{10}$$

Eq. (10) can be solved in polynomial time using the min-cut algorithm proposed in [3]. We then perform the bound check on the current guess g based on Theorem 1.

Theorem 1. *Assume subgraph $\mathcal{A}_s^*$ and $\mathcal{A}_t^*$ give the minimum cut c^* and subgraph $\mathcal{A}^*$ solves Eq. (1) and (3). If $c^* > m \times |\mathcal{T}| - 2 \times g \times \eta$, then $g > D(\mathcal{A}^*)$; if $c^* < m \times |\mathcal{T}| - 2 \times g \times \eta$, then $g < D(\mathcal{A}^*)$; if $c^* = m \times |\mathcal{T}| - 2 \times g \times \eta$, then $g = D(\mathcal{A}_s^*)$ and $D(\mathcal{A}_s^*) = D(\mathcal{A}^*)$.*

Proof. Notice that if $c^* > m \times |\mathcal{T}| - 2 \times g \times \eta$, then $g > D(\mathcal{A}) \ \forall \mathcal{A} \subseteq \mathcal{T}$ based on Eq. (9) since c^* is the minimum cut capacity. We have g as an upper bound of the optimal density $D(\mathcal{A}^*)$. If $c^* < m \times |\mathcal{T}| - 2 \times g \times \eta$, then $\exists \mathcal{A} \subseteq \mathcal{T}$ such that $g < D(\mathcal{A})$ based on Eq. (9) since c^* is the minimum cut capacity. We have g as an lower bound of the optimal density $D(\mathcal{A}^*)$. If $c^* = m \times |\mathcal{T}| - 2 \times g \times \eta$, then $g \geq D(\mathcal{A}) \ \forall \mathcal{A} \subseteq \mathcal{T}$ and $g = D(\mathcal{A}_s^*)$ based on Eq. (9) since c^* is the minimum cut capacity. Hence, $D(\mathcal{A}_s^*) \geq D(\mathcal{A}) \forall \mathcal{A} \subseteq \mathcal{T}$ and $D(\mathcal{A}_s^*) = D(\mathcal{A}^*)$. □

Algorithm: Based on Theorem 1, we implement our binary search strategy to iteratively shrink the lower and upper bound of the optimal density $D(\mathcal{A}^*)$. However, since both our node and edge weights are continuous values, $D(\mathcal{A})$ is also continuous for $\mathcal{A} \subseteq \mathcal{T}$. We have to search infinitely to find $\mathcal{A}^*$. In practice, we can specify an error bound to stop the search. For example, let $\hat{\mathcal{A}}$ be our candidate solution, u and l be the current upper and lower bound. If a solution satisfying $\frac{D(\mathcal{A}^*) - D(\hat{\mathcal{A}})}{D(\mathcal{A}^*)} \leq \beta$ is good enough, we can safely stop the search when $\frac{u-l}{l} \leq \beta$. A tighter bound needs more iterations. The entire algorithm is summarized in Algorithm 1. Note that, we update the candidate subgraph $\hat{\mathcal{A}}$ to $\mathcal{A}_s^*$ when g is the determined to be the new lower bound because $D(\mathcal{A}_s^*)$ is greater than the new lower bound g and smaller than the current upper bound u. This is not true when g is the upper bound as $D(\mathcal{A}_s^*)$ may be smaller than the current lower bound. The initial lower bound is set to the maximum η-density of any 2-node subgraph of $\mathbb{G}$ to avoid zero lower bound, and the initial upper bound is set to the summation of all the node and edge weights in the graph. A loose bound like this does not affect the efficiency of our algorithm as binary search shrinks the bounds exponentially. Let U denote the initial upper bound, we need to perform $O(log(\frac{U}{\beta \times D(\mathcal{A}^*)}))$ graph cut operations throughout the algorithm. It is logarithmic in terms of the problem size.

Simulation Experiment: In order to visualize the effectiveness of the proposed density maximization approach, a test experiment is performed on simulated 2D data points. These data points are drawn from three 2D Gaussian distributions and one 2D uniform distribution, as shown in the first plot of Figure 4. In this simulation experiment, the points' unary scores are set to zero as they are difficult to visualize. The affinity graph is constructed in the same way as described in Section 3.1. The selection results using the proposed method are shown in the subsequent plots of Figure 4. It can be seen that, the selection is quite conservative when $\eta = 0$. It completely misses the two dense modes at

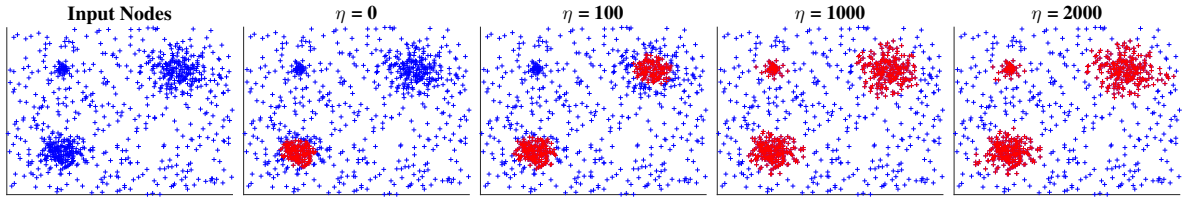

Figure 4. Dense subgraph selection results in the simulation experiment. The red dots denote the selected nodes under different η settings.

Algorithm 1 Maximum η-Density Optimization

Input: Graph $G = (\mathcal{T}, \mathcal{E})$, error bound β
Output: Subgraph $\hat{\mathcal{A}} \subseteq \mathcal{T}$ achieving the maximum η-density defined by Eq. (3) within the given error bound
$l \leftarrow \frac{\max_{t_i^j \in \mathcal{T}, t_p^q \in \mathcal{T}, t_i^j \neq t_p^q} s_i^j + s_p^q + w(t_i^j, t_p^q)}{\eta + 2}$
$u \leftarrow \sum_{t_i^j \in \mathcal{T}} s_i^j + \sum_{(t_i^j, t_p^q) \in \mathcal{E}} w(t_i^j, t_p^q)$
$\hat{\mathcal{A}} \leftarrow$ the two-node subgraph achieving l
while $\frac{u-l}{l} > \beta$ **do**
    $g \leftarrow \frac{u+l}{2}$
    Find c^*, $\mathcal{A}_s^*$ and $\mathcal{A}_t^*$ in Eq. (10) by max flow [3].
    if $c^* > m \times |\mathcal{T}| - 2 \times g \times \eta$ **then**
        $u \leftarrow g$
    else if $c^* < m \times |\mathcal{T}| - 2 \times g \times \eta$ **then**
        $\hat{\mathcal{A}} \leftarrow \mathcal{A}_s^*$
        $l \leftarrow g$
    else
        $\hat{\mathcal{A}} \leftarrow \mathcal{A}_s^*$
        break
    end if
end while

the top as well as the outer region of the mode at the bottom. A larger η relaxes the strict average regularization and the algorithm selects all three modes. It is worth noting that the selection does not change much when we increase η from 1000 to 2000, which indicates the proposed method is not particularly sensitive to η at this range.

3.3. Action Proposal Generation and Description

Compared with generic video object proposals [15, 26, 36] or motion based action proposals [37], human prior has shown to be a much more accurate cue to detect human actions [20, 41, 42]. In this work, we also start with per-frame human detections to build spatio-temporal action proposals. The Faster R-CNN (VGG16) object detection framework [28] is used here for its good performance. However, a critical drawback of human detection is that it usually misses the humans undergoing severe pose variations or occlusions while performing an action. Hence, to improve the recall, we fuse per-frame detection results of two Faster-RCNN models. One is trained on the VOC2007 [10] train-validation subset containing daily human photos with moderate pose variation, and the other is trained on the MPII human activity dataset [41] containing human photos with large pose variations. Spatio-temporal action detection proposals are then generated by linking the per-frame human detections using the method proposed in [42]. Furthermore, since the linking process does not produce new human detections, we also apply tracking to generate more spatio-temporal action proposals to improve the recall [41].

After extracting spatio-temporal action proposals, we break each proposal tube into a sequence of tubelets with 16 frames long and 8 frames overlap. These tubelets are then described by the 4096-dimensional fc-6 activations of the C3D network [34] trained on Sport-1M dataset [19]. A proposal tube's feature vector is computed as the average of all its tubelets' C3D features followed by ℓ_2 normalization. PCA is also used to reduce the feature dimensions to 512.

4. Experiments

4.1. Datasets

To evaluate the proposed common action co-localization method in unconstrained scenarios, we first build two datasets. In both datasets, some videos contain one or multiple common actions, while some videos contain no common actions, *i.e.*, outlier videos. Many outlier videos also contain actions but they are not common in the dataset.

The first dataset is the *UCF Sports Plus* dataset. It includes all the 150 videos (10 action classes) in the *UCF Sports Action* [29] dataset as common actions. We re-annotate the videos containing multiple common actions to include them all. We also add 70 outlier videos containing no common actions. The second dataset is the *SVW Mini* dataset. It includes the annotated bowling and golfing videos in the SVW (Sports Action in the Wild) [30] dataset as common actions. Besides adding the 70 outlier videos in the *UCF Sports Plus* dataset, we also pick one video from each of the rest action classes in the SVW dataset as additional outlier videos. In total, there are 216 videos in this dataset, 120 of which contain common actions. Some example frames in these two datasets are shown in Figure 5.

4.2. Evaluation Criteria

Similar to previous video object co-localization works [16, 18, 21], our co-localization method returns a ranked

Figure 5. Sample frames in the newly proposed *UCF Sports Plus* (left two columns) and *SVW Mini* (right two columns) datasets. The bounding boxes denote the common action annotations.

list of localizations for each video. A ground truth is recalled if its intersection over union (IOU) ratio with a localization is greater than a threshold. Most previous works have evaluated the co-localization performance separately for each video, *i.e.*, flag a video as correctly localized if the common object in the video is covered by the top 1 detection. While this metric is meaningful in the constrained case where each video contains exactly one common object, it is not suitable in our unconstrained action co-localization scenario where a video may contain zero or multiple common actions. Indeed, this evaluation criterion implicitly excludes all the outlier videos containing no common actions. Thus, in this work, we use a different evaluation criterion to test the performance. We first put all the detections, including outlier videos, in a single ranked listed, and then compute the precision-recall curve and average precision to evaluate the localization performance. Note that, a ground truth common action can only be recalled once and all subsequent detections are treated as false positives.

4.3. Comparison with Baselines

In this section, we compare with several baselines and other subgraph selection methods to validate the advantage of the proposed co-localization method. In the following description, we use K to denote the actual number of common action classes in the dataset.

- Select all proposals and assign them random scores.
- Select all proposals and use their original scores.
- Set $\eta = 0$, *i.e.*, the average degree subgraph density formulation proposed in [13].
- Use the graph cut based subgraph selection formulation proposed in [2].
- Use the Cohesive Subgraph Mining (CSGM) formulation proposed in [48].
- Use K-Means to cluster the proposals into $K+1$ clusters, and remove the cluster with most number of outlier tubes.
- Use the Dominant Set clustering method with the proposed pealing off strategy in [27] to cluster the proposals and remove those un-clustered proposals.
- Select all proposals after excluding the videos containing no common actions.

The precision recall curves of these methods are shown in Figure 6, and the average precisions are shown in Table 1. We use a lower threshold, *i.e.*, 0.25, for *SVW Mini* as

Table 1. The average precisions of the proposed method as well as several baselines on the action co-localization task.

	UCF Sports Plus		*SVW Mini*
	IOU = 0.5	IOU = 0.25	IOU = 0.25
random	8.67%	21.73%	7.78%
original	31.27%	35.62%	25.60%
$\eta = 0$ [13]	5.88%	7.41%	0.83%
graph cut [2]	31.28%	35.63%	28.83%
k-means	32.24%	37.99%	33.83%
CSGM [48]	31.38%	35.82%	25.68%
DomSet [27]	41.79%	49.65%	46.75%
ours	**50.29%**	**58.49%**	**48.17%**
w/o outliers	71.23%	80.41%	71.30%

Table 3. Clustering accuracies (F-Measures) on the original proposals and our selected proposals on the *SVW Mini* dataset.

	bowling	golf	average
original	0.20	0.15	0.17
ours	0.43	0.22	0.32
w/o outlier videos	0.34	0.25	0.30
w/o outlier tubes	0.96	0.97	0.97

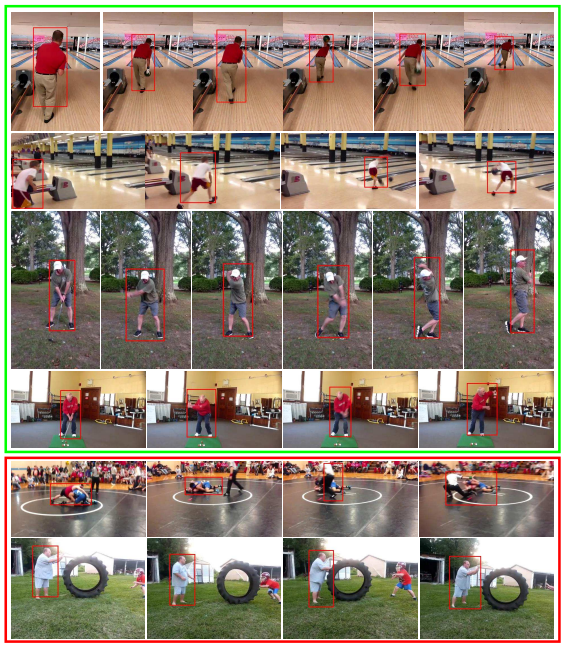

Figure 7. Although all the shown proposals capture valid human actions, some are selected (top four rows) by our method as they contain common actions in the dataset, while some are rejected (bottom two rows) as they contain non-common actions.

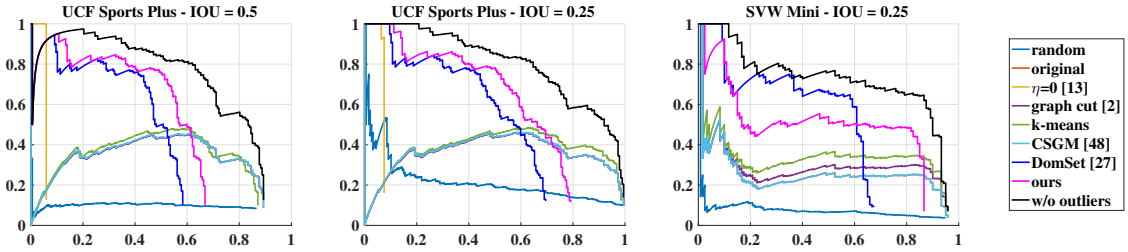

Figure 6. Precision recall curves of the proposed method as well as several baselines on the action co-localization task. The curve for "original" is not quite visible because it largely overlaps with the "graph cut" or "CSGM" curve.

Table 2. Clustering accuracies (F-Measures) on the original proposals and our selected proposals on the *UCF Sports Plus* dataset.

	dive	golf	kick	lift	horse ride	run	skateboard	swing	angle swing	walk	average
original	0.55	0.048	0.22	0.19	0.39	0.35	0.036	0.57	0.35	0.26	0.29
ours	0.62	0.074	0.11	0.50	0.46	0.38	0.053	0.67	0.46	0.33	0.37
w/o outlier videos	0.61	0.20	0.26	0.36	0.63	0.39	0.014	0.62	0.42	0.38	0.39
w/o outlier tubes	0.92	0.32	0.52	0.78	0.82	0.60	0.33	0.87	0.90	0.53	0.66

its ground truth annotation is loose, as shown in Figure 5. Note that, if we manually exclude the outlier videos, *i.e.*, the method "w/o outliers", we can achieve a high average precision just using the original proposals. This demonstrates the good quality of the initial action proposals. After adding outlier videos, *i.e.*, the method "original", the average precision drops significantly. This shows the importance of the proposed action co-localization problem in unconstrained scenarios. The proposed method successfully selects the proposals containing common actions, and improves the average precision by more than 20% percent in all the datasets and IOU settings. It is worth noting that, when $\eta = 0$, the average precision is even lower than the random case but the initial precision is almost perfect in the first two precision recall curves. The lower average precision is due to the extremely low recall as the selection is too strict. This further demonstrates the importance of a relaxed regularization. The graph cut based formulation [2] has only marginally improved the original baseline. This is because the graph cut formulation only minimizes the edge weights between selected and unselected nodes, while does not enforce strong edges within the selected nodes. Hence, this formulation is useful only when the node weights are mostly reliable, which is not true in our case due to the existence of "non-common" actions. The method "K-Means" does not perform well because it assumes the outlier proposals can also form a compact cluster. The "CSGM" method is not performing well as their formulation can only be solved approximately [48]. The method "DomSet" is the best among the compared methods but still outperformed by ours, especially on the *UCF Sports Plus* dataset where its PR curve is consistently lower than ours. On *SVW Mini*, it has a better initial precision but the final recall is lower. In addition, "DomSet" is much slower than ours in our experiment.

Some qualitative results are shown in Figure 7 to demonstrate how our method successfully selects proposals containing common actions and rejects proposals containing uncommon actions. We also show some actual action co-localization results in Figure 8. It is worth noting that, in the top left row of Figure 8, there are actually two humans. Indeed, the proposal highlighting the right person has a higher proposal confidence score, but our method selects the left person as there are many golfing actions in this dataset.

To further demonstrate the effectiveness of the proposed co-localization approach. We perform K-Means to cluster the selected proposals into $K + 1$ clusters. We assign each cluster an action label based on majority voting and compute the precision, recall, and f-measure of each action class. The comparisons before and after the node selection are shown in Table 2 and 3. The method "w/o outlier videos" means we manually exclude the videos containing no common actions. The method "w/o outlier tubes" means we manually exclude the proposal tubes capturing no common actions. It can be seen that, the proposed method apparently improves the baseline and are comparable to the method "w/o outlier videos".

In order to show our method can also generalize to larger video sets. We perform extra experiments on the JHMDB [17] and UCF101 (test set of the 24-class detection subset) [32] datasets with added outlier videos. In total, there are 1010 videos (45K frames), and 997 videos (200K frames), respectively. The mAP for the input proposals, [13] and our method are 64.83%, 1.94%, 69.00% for JHMDB with an IOU threshold of 0.5, and 26.07%, 3.43%, 37.92% for UCF101 with an IOU threshold of 0.25. This shows that our method can improve the baselines in these larger sets.

4.4. Comparison with Video Object Co-localization

In this section, we compare with [16] to show that it is non-trivial to apply existing video object co-localization methods to common action co-localizations. In [16], it generates one localization for each video (including out-

Figure 8. Our action co-localization results on the *UCF Sports Plus* (left) and *SVW Mini* (right) datasets. The top rows in the green blocks contain videos with common actions. The bottom rows in the red blocks contain videos that do not have common actions. It can be seen that, the proposed method can handle the cases when there are zero, one or multiple common actions in the videos.

Table 4. Comparison with [16] using correct detection ratio metric.

	[16]	ours
UCF Sports Plus (IoU=0.5)	31.85%	49.20%

Table 5. The running time of each step in the proposed co-localization approach on a dataset containing 23013 frames.

	time	% of total time
human detection	200 min	48.78%
action proposal generation	130 min	31.70%
feature extraction + PCA	79 min	19.27%
affinity graph construction	6.1 sec	0.025%
subgraph selection	1.2 sec	0.0049%
total	410 min	-

lier videos) because it assumes each video contains exactly one common object and the detection scores between different videos are not comparable. For a fair comparison, we use [16]'s correct detection ratio metric, exclude all outlier videos and only use the top 1 detection of our method in each video. The comparison results are shown in Table 4. It can be seen that our method produces more accurate localization results. Furthermore, [16] cannot identify outlier videos due to their assumption.

4.5. Running Time

The detailed running time of the proposed co-localization approach on the *UCF Sports Plus* dataset is shown in Table 5. For this dataset, there are 23013 frames in total and the affinity graph contains 2142 nodes and 41102 edges. Furthermore, for the larger UCF101 dataset with 200K frames, affinity graph construction and common proposal selection take 138 and 1.1 seconds, respectively. It can be seen that most of the time is spent on the proposal generation and feature extraction steps, while the proposed optimal subgraph selection algorithm is efficient.

4.6. Limitations and Future Work

It is worth noting that, due to human prior, the proposal generation method we use cannot handle untrimmed case where a human performs common action only during part of his/her presence. However, the proposed common action selection method still has great potential of handling untrimmed videos as long as we can obtain reasonable action proposals. In the future, we will consider to propose more robust action proposals to handle untrimmed videos. We will also explore the potential of our selection method to directly refine temporally untrimmed proposals.

5. Conclusion

In this work, we tackle the problem of automatic common action discovery and localization in unconstrained videos. We are unaware of which types of action are common, and each video may contain zero, one or several common action instances. In the proposed method, we first generate action proposals and then select the proposals containing common actions by solving a subgraph density maximization problem. A polynomial time algorithm is also proposed to solve it. The evaluation results on several datasets demonstrate the effectiveness of the proposed method.

Acknowledgment

This work is supported in part by Singapore Ministry of Education Academic Research Fund Tier 2 MOE2015-T2-2-114 and Tier 1 RG27/14. This research was carried out at Rapid-Rich Object Search (ROSE) Lab at Nanyang Technological University, Singapore. ROSE Lab is supported by the National Research Foundation, Singapore, under its Interactive Digital Media (IDM) Strategic Research Programme. We gratefully acknowledge the support of NVIDIA AI Tech Centre for our research at ROSE Lab.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *T-PAMI*, 34(11):2274–2282, 2012.

[2] S. D. Bhattacharjee, J. Yuan, Y.-P. Tan, and L.-Y. Duan. Query-adaptive small object search using object proposals and shape-aware descriptors. *T-MM*, 18(4):726–737, 2016.

[3] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *T-PAMI*, 26(9):1124–1137, 2004.

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

[5] W.-S. Chu, F. Zhou, and F. De la Torre. Unsupervised temporal commonality discovery. In *ECCV*, pages 373–387. Springer, 2012.

[6] L. Del Pero, S. Ricco, R. Sukthankar, and V. Ferrari. Articulated motion discovery using pairs of trajectories. In *CVPR*, pages 2151–2160, 2015.

[7] L. Del Pero, S. Ricco, R. Sukthankar, and V. Ferrari. Behavior discovery and alignment of articulated object classes from unstructured video. *IJCV*, pages 1–23, 2016.

[8] L. Del Pero, S. Ricco, R. Sukthankar, and V. Ferrari. Discovering the physical parts of an articulated object class from multiple videos. In *CVPR*, pages 714–723, 2016.

[9] I. Endres and D. Hoiem. Category independent object proposals. *Computer Vision–ECCV 2010*, pages 575–588, 2010.

[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[11] H. Fu, D. Xu, B. Zhang, and S. Lin. Object-based multiple foreground video co-segmentation. In *CVPR*, pages 3166–3173. IEEE, 2014.

[12] H. Fu, D. Xu, B. Zhang, S. Lin, and R. K. Ward. Object-based multiple foreground video co-segmentation via multi-state selection graph. *T-IP*, 24(11):3415–3424, 2015.

[13] A. V. Goldberg. Finding a maximum density subgraph. Technical report, University of California at Berkeley, Berkeley, CA, USA, 1984.

[14] J. Guo, Z. Li, L.-F. Cheong, and S. Zhiying Zhou. Video co-segmentation for meaningful action extraction. In *ICCV*, pages 2232–2239, 2013.

[15] M. Jain, J. Van Gemert, H. Jégou, P. Bouthemy, and C. G. Snoek. Action localization with tubelets from motion. In *CVPR*, pages 740–747, 2014.

[16] K. R. Jerripothula, J. Cai, and J. Yuan. Cats: Co-saliency activated tracklet selection for video co-localization. In *ECCV*, pages 187–202. Springer, 2016.

[17] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *International Conf. on Computer Vision (ICCV)*, pages 3192–3199, Dec. 2013.

[18] A. Joulin, K. Tang, and L. Fei-Fei. Efficient image and video co-localization with frank-wolfe algorithm. In *ECCV*, pages 253–268. Springer, 2014.

[19] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732, 2014.

[20] A. Kläser, M. Marszałek, C. Schmid, and A. Zisserman. Human focused action localization in video. In *ECCV*, pages 219–233. Springer, 2010.

[21] S. Kwak, M. Cho, I. Laptev, J. Ponce, and C. Schmid. Unsupervised object discovery and tracking in video collections. In *ICCV*, pages 3173–3181, 2015.

[22] J. Liu and Y. Liu. Grasp recurring patterns from a single view. In *CVPR*, pages 2003–2010, 2013.

[23] J. Meng, H. Wang, J. Yuan, and Y.-P. Tan. From keyframes to key objects: Video summarization by representative object proposal selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1039–1048, 2016.

[24] P. Mettes, J. C. van Gemert, and C. G. Snoek. Spot on: Action localization from pointly-supervised proposals. *arXiv preprint arXiv:1604.07602*, 2016.

[25] E. A. Mosabbeb, R. Cabral, F. De la Torre, and M. Fathy. Multi-label discriminative weakly-supervised human activity recognition and localization. In *ACCV*, pages 241–258. Springer, 2014.

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

[27] M. Pavan and M. Pelillo. Dominant sets and pairwise clustering. *T-PAMI*, 29(1), 2007.

[28] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.

[29] M. D. Rodriguez, J. Ahmed, and M. Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, pages 1–8. IEEE, 2008.

[30] S. M. Safdarnejad, X. Liu, L. Udpa, B. Andrus, J. Wood, and D. Craven. Sports videos in the wild (svw): A video dataset for sports analysis. In *FG*, volume 1, pages 1–7. IEEE, 2015.

[31] P. Siva and T. Xiang. Weakly supervised action detection. In *BMVC*, volume 2, page 6, 2011.

[32] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[33] K. Tang, A. Joulin, L.-J. Li, and L. Fei-Fei. Co-localization in real-world images. In *CVPR*, pages 1464–1471. IEEE, 2014.

[34] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497. IEEE, 2015.

[35] Y.-H. Tsai, G. Zhong, and M.-H. Yang. Semantic co-segmentation in videos. In *ECCV*, pages 760–775. Springer, 2016.

[36] Z. Tu, Z. Guo, W. Xie, M. Yan, R. C. Veltkamp, B. Li, and J. Yuan. Fusing disparate object signatures for salient object detection in video. *Pattern Recognition*, 2017.

[37] J. van Gemert, M. Jain, E. Gati, and C. Snoek. Apt: Action localization proposals from dense trajectories. In *BMVC*, volume 2, page 4, 2015.

[38] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[39] L. Wang, G. Hua, R. Sukthankar, J. Xue, Z. Niu, and N. Zheng. Video object discovery and co-segmentation with weak supervision. *T-PAMI*, 2016.

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

[41] P. Weinzaepfel, X. Martin, and C. Schmid. Towards weakly-supervised action localization. *arXiv preprint arXiv:1605.05197*, 2016.

[42] G. Yu and J. Yuan. Fast action proposals for human action detection and search. In *CVPR*, pages 1302–1311, 2015.

[43] G. Yu, J. Yuan, and Z. Liu. Unsupervised random forest indexing for fast action search. In *CVPR*, pages 865–872. IEEE, 2011.

[44] G. Yu, J. Yuan, and Z. Liu. Action search by example using randomized visual vocabularies. *T-IP*, 22(1):377–390, 2013.

[45] J. Yuan, J. Meng, Y. Wu, and J. Luo. Mining recurring events through forest growing. *T-CSVT*, 18(11):1597–1607, 2008.

[46] J. Yuan and Y. Wu. Spatial random partition for common visual pattern discovery. In *ICCV*, pages 1–8. IEEE, 2007.

[47] D. Zhang, O. Javed, and M. Shah. Video object co-segmentation by regulated maximum weight cliques. In *ECCV*, pages 551–566. Springer, 2014.

[48] G. Zhao and J. Yuan. Discovering thematic patterns in videos via cohesive sub-graph mining. In *ICDM*, pages 1260–1265. IEEE, 2011.

[49] G. Zhao, J. Yuan, J. Xu, and Y. Wu. Discovering the thematic object in commercial videos. *T-MM*, 18(3):56–65, 2011.