Common Action Discovery and Localization in Unconstrained Videos

Jiong Yang  Junsong Yuan
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 viewpoint 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.

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 $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ 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, $v^i_j$ denote the node
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 \( \{t^l_i\} \) and their feature descriptions \( \{f^l_i\} \), we build an \( \epsilon \)-neighborhood affinity graph, \( G = \langle T, E \rangle \), using all the proposal nodes. Node \( t^l_i \) and \( t^q_j \) will be linked only if \( \|f^l_i - f^q_j\|_2 \leq \epsilon \), where \( \| \cdot \|_2 \) denotes the \( \ell_2 \) distance and \( \epsilon \) is the bandwidth for graph construction. The edge weight \( w(t^l_i, t^q_j) \) is computed as

\[
w(t^l_i, t^q_j) = \exp\left(-\frac{\|f^l_i - f^q_j\|_2^2}{2 \times \beta^2}\right),
\]

where \( \beta \) is computed as

\[
\beta = \frac{\sum_{(t^l_i, t^q_j) \in E} \|f^l_i - f^q_j\|_2}{|E|},
\]

and \( |E| \) 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 \( \eta \) 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 \( \eta \) values. The proposed method uses a binary search strategy to find the optimal density \( D(A^*) \), i.e., given the current lower bound \( l \) and upper bound \( u \) on \( D(A^*) \), we first check if our new guess \( g = \frac{u + l}{2} \) defines a lower or upper bound on \( D(A^*) \) and then shrink the search space by half. A candidate subgraph whose \( \eta \)-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 \( \lambda \) in Eq. (3) without loss of generality.

**Bound Check:** To perform the bound check on our current guess \( g \) on \( D(A^*) \), 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 \( d^l_i \) denote the degree of node \( t^l_i \), i.e., the weight summation of all the edges connected to \( t^l_i \), the newly added source to node weight \( \omega(s, t^l_i) \) and node to sink weight \( \omega(t^l_i, t) \) are defined as

\[
w(s, t^l_i) = m,
\]

\[
w(t^l_i, t) = m + 2 \times g - d^l_i - 2 \times s^l_i,
\]
Interestingly, the summation of the edge weights along the cut boundary.

\[ c(A, s, t) = m + 2xg - d_i^j - 2xs_i^j \]

where the variable \( m \) is defined as

\[ m = \max_{t_i^j \in T} (d_i^j + 2xs_i^j). \quad (8) \]

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 \( A_s \) and \( A_t \), respectively. The cut capacity \( c(A_s, A_t) \) is defined as the summation of the edge weights along the cut boundary. Interestingly, \( c(A_s, A_t) \) is related to the \( \eta \)-density of subgraph \( A_s \), i.e., \( D(A_s) \), as follows:

\[
c(A_s, A_t) = \sum_{t_i^j \in A_t} w(s, t_i^j) + \sum_{t_i^j \in A_s} w(t_i^j, t) + \sum_{t_i^j \in A_s, t_i^j \in A_t} w(t_i^j, t_i^j) = m \times |A_t| + \left( m \times |A_s| + 2xg \times |A_s| \right) - \sum_{t_i^j \in A_s} d_i^j - 2 \sum_{t_i^j \in A_s, t_i^j \in A_t} s_i^j - \sum_{t_i^j \in A_s, t_i^j \in A_t} w(t_i^j, t_i^j) = m \times |A_t| + 2xg \times |A_s| - 2xg \times \eta + 2 \sum_{t_i^j \in A_t} \eta \times w(t_i^j, t_i^j) = m \times |A_t| - 2xg \times \eta + 2 \times (|A_s| + \eta) \times (g - D(A_s)) \]

\[
\frac{D(A_s) - D(A_t)}{D(A_s)} \leq \beta \text{ is good enough, we can safely stop the search when } \frac{D(A_s) - D(A_t)}{D(A_s)} \leq \beta. \quad \text{Eq. (9)}
\]

Algorithm: Based on Theorem 1, we implement our binary search strategy to iteratively shrink the lower and upper bound of the optimal density \( D(A_s) \). However, since both our node and edge weights are continuous values, \( D(A) \) is also continuous for \( A \subseteq T \). We have to search infinitely to find \( A^* \). In practice, we can specify an error bound to stop the search. For example, let \( A \) be our candidate solution, \( u \) and \( l \) be the current upper and lower bound. If a solution satisfying \( D(A) \) is greater than the new lower bound

\[ D(A) = \min_{A \subseteq T} \left\{ D(A) \mid \eta \right\} \]

is greater than the new lower bound

\[ D(A) > \min_{A \subseteq T} \left\{ D(A) \mid \eta \right\} \]

then \( \eta \) completely misses the two dense modes at

Theorem 1. Assume subgraph \( A_s^* \) and \( A_t^* \) give the minimum cut \( c^* \) and subgraph \( A_s^* \) solves Eq. (1) and (3). If \( c^* > m \times |T| - 2xg \times \eta \), then \( g > D(A_s^*) \); if \( c^* < m \times |T| - 2xg \times \eta \), then \( g < D(A_s^*) \); if \( c^* = m \times |T| - 2xg \times \eta \), then \( g = D(A_s^*) \) and \( D(A_s) = D(A_s^*) \).

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

Algorithm: Based on Theorem 1, we implement our binary search strategy to iteratively shrink the lower and upper bound of the optimal density \( D(A_s^*) \). However, since both our node and edge weights are continuous values, \( D(A) \) is also continuous for \( A \subseteq T \). We have to search infinitely to find \( A^* \). In practice, we can specify an error bound to stop the search. For example, let \( A \) be our candidate solution, \( u \) and \( l \) be the current upper and lower bound. If a solution satisfying \( D(A) \) is greater than the new lower bound

\[ D(A) = \min_{A \subseteq T} \left\{ D(A) \mid \eta \right\} \]

is greater than the new lower bound

\[ D(A) > \min_{A \subseteq T} \left\{ D(A) \mid \eta \right\} \]

then \( \eta \) completely misses the two dense modes at

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
Algorithm 1 Maximum \( \eta \)-Density Optimization

1. **Input:** Graph \( G = (\mathcal{T}, \mathcal{E}) \), error bound \( \beta \)
2. **Output:** Subgraph \( \mathcal{A} \subseteq \mathcal{T} \) achieving the maximum \( \eta \)-density defined by Eq. (3) within the given error bound
3. \( l \leftarrow \max_{i, j \in \mathcal{T}} \eta \frac{\sum_{i \in \mathcal{T}} s^1_i + \sum_{i \in \mathcal{T}} w(t_i^j, t_j^i)}{\eta + 2} \)
4. \( u \leftarrow \sum_{i \in \mathcal{T}} s^1_i + \sum_{i \in \mathcal{T}} w(t_i^j, t_j^i) \)
5. \( \mathcal{A} \leftarrow \) the two-node subgraph achieving \( l \)
6. while \( \frac{u - l}{u} > \beta \) do
7. \( g \leftarrow \frac{u - l}{2} \)
8. Find \( c^*, A^* \) and \( A^* \) in Eq. (10) by max flow [3].
9. if \( c^* > m \times |\mathcal{T}| - 2\times g \times \eta \) then
10. \( u \leftarrow g \)
11. else if \( c^* < m \times |\mathcal{T}| - 2\times g \times \eta \) then
12. \( \mathcal{A} \leftarrow A^* \)
13. \( l \leftarrow g \)
14. else
15. \( \mathcal{A} \leftarrow A^* \)
16. break
17. end if
18. end while

the top as well as the outer region of the mode at the bottom. A larger \( \eta \) 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 \( \eta \) from 1000 to 2000, which indicates the proposed method is not particularly sensitive to \( \eta \) 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] 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 \( \ell_2 \) 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
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 list, 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 peeling 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>
<thead>
<tr>
<th>Method</th>
<th>UCF Sports Plus (IOU = 0.5)</th>
<th>SVW Mini (IOU = 0.25)</th>
</tr>
</thead>
<tbody>
<tr>
<td>random</td>
<td>8.67%</td>
<td>7.78%</td>
</tr>
<tr>
<td>original</td>
<td>31.27%</td>
<td>25.60%</td>
</tr>
<tr>
<td>$\eta = 0$ [13]</td>
<td>5.88%</td>
<td>0.83%</td>
</tr>
<tr>
<td>graph cut [2]</td>
<td>31.28%</td>
<td>28.83%</td>
</tr>
<tr>
<td>k-means</td>
<td>32.24%</td>
<td>33.83%</td>
</tr>
<tr>
<td>CSGM [48]</td>
<td>31.38%</td>
<td>25.68%</td>
</tr>
<tr>
<td>DomSet [27]</td>
<td>41.79%</td>
<td>46.75%</td>
</tr>
<tr>
<td>ours</td>
<td>50.29%</td>
<td>48.17%</td>
</tr>
<tr>
<td>w/o outliers</td>
<td>71.23%</td>
<td>71.30%</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Method</th>
<th>bowling</th>
<th>golf</th>
<th>average</th>
</tr>
</thead>
<tbody>
<tr>
<td>original</td>
<td>0.20</td>
<td>0.15</td>
<td>0.17</td>
</tr>
<tr>
<td>ours</td>
<td>0.43</td>
<td>0.22</td>
<td>0.32</td>
</tr>
<tr>
<td>w/o outlier videos</td>
<td>0.34</td>
<td>0.25</td>
<td>0.30</td>
</tr>
<tr>
<td>w/o outlier tubes</td>
<td>0.96</td>
<td>0.97</td>
<td>0.97</td>
</tr>
</tbody>
</table>

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

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

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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>[16]</th>
<th>ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCF Sports Plus (IoU=0.5)</td>
<td>31.85%</td>
<td>49.20%</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Step</th>
<th>time</th>
<th>% of total time</th>
</tr>
</thead>
<tbody>
<tr>
<td>human detection</td>
<td>200 min</td>
<td>48.78%</td>
</tr>
<tr>
<td>action proposal generation</td>
<td>130 min</td>
<td>31.70%</td>
</tr>
<tr>
<td>feature extraction + PCA</td>
<td>79 min</td>
<td>19.27%</td>
</tr>
<tr>
<td>affinity graph construction</td>
<td>6.1 sec</td>
<td>0.025%</td>
</tr>
<tr>
<td>subgraph selection</td>
<td>1.2 sec</td>
<td>0.0049%</td>
</tr>
<tr>
<td>total</td>
<td>410 min</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 4 shows the comparison with [16] using the correct detection ratio metric. 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


